

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA QUÍMICA

DALVA JANINE RITA

**CONTROLE DE PROCESSOS USANDO
REDES NEURAIS ARTIFICIAIS:
UMA APLICAÇÃO EXPERIMENTAL**

Orientador:

Prof^o Dr. Ariovaldo Bolzan

Co-orientador:

Prof^o Dr. Leonel Teixeira Pinto

Dissertação apresentada ao Curso de Pós-Graduação em Engenharia Química do Centro Tecnológico da Universidade Federal de Santa Catarina, como requisito parcial à obtenção do Título de Mestre em Engenharia Química.

Florianópolis - SC - 1995

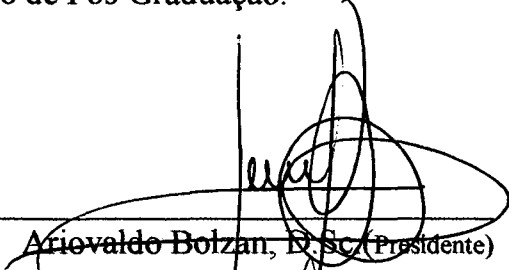
**Controle de Processos Usando Redes Neurais Artificiais:
Uma Aplicação Experimental**

por

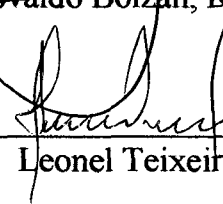
Dalva Janine Rita

Esta dissertação foi julgada para a obtenção do título de
Mestre em Engenharia
Especialidade **Engenharia Química**
Área de Concentração **Desenvolvimento de Processos Químicos e Biotecnológicos**
e aprovada em sua forma final pelo curso de Pós-Graduação.

Banca Examinadora:



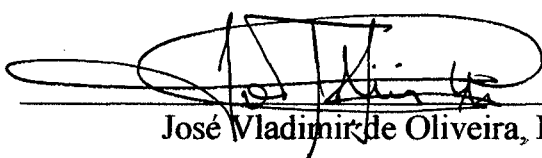
Ariovaldo Bolzan, D.Sc. (Presidente)



Leonel Teixeira Pinto, D.Sc.



Enrique Luis Lima, D. Sc.



José Vladimir de Oliveira, D. Sc.

Florianópolis, 30 de agosto de 1995.

AGRADECIMENTOS

- Ao Prof^o Dr. Ariovaldo Bolzan pela orientação e principalmente pela amizade e compreensão.
- Ao Prof^o Dr. Leonel Teixeira Pinto pelas valiosas sugestões.
- Aos colegas do Laboratório de Controle de Processos pela amizade e contribuição na realização deste trabalho.
- Ao Serv. Edevilson Silva por ter sido sempre tão prestativo, competente e amigo.
- Ao CNPq pelo apoio financeiro.
- Aos meus pais, Amaro, Izabel, Guedes e Conceição, pelo incentivo.

*'Os seus amigos o conhecerão melhor no primeiro
minuto em que se conhecerem do que os seus conhecidos
o conhecerão em mil anos.'*

Richard Bach

Aos meus verdadeiros amigos

SUMÁRIO

ÍNDICE DE TABELAS	i
ÍNDICE DE FIGURAS	ii
RESUMO	iv
SUMMARY	v
NOMENCLATURA	vi
CAPÍTULO I - INTRODUÇÃO	1
CAPÍTULO II - REVISÃO BIBLIOGRÁFICA	4
II.1 Redes Neurais Artificiais - uma definição geral.....	4
II.2 Redes Neurais Artificiais - um breve histórico.....	4
II.3 Aplicações de Redes Neurais Artificiais na Engenharia Química.....	7
CAPÍTULO III - FUNDAMENTOS TEÓRICOS	12
III.1 Redes Neurais Artificiais.....	12
III.1.1 Arquitetura das Redes Neurais Artificiais.....	13
III.1.2 Os neurônios artificiais.....	14
III.2 O método de aprendizagem.....	17
III.2.1 Definição e equações.....	17
III.2.2 O algoritmo <i>backpropagation</i>	20
III.2.3 Utilização da rede após o treinamento.....	21
III.3 Implementação de Redes Neurais Artificiais.....	22
III.4 Vantagens e desvantagens de Redes Neurais Artificiais.....	23
III.5 Modelagem de processos utilizando Redes Neurais Artificiais.....	25
III.6 Controle de Processos baseado em Redes Neurais Artificiais.....	27
III.6.1 Estratégias de controle usando Redes Neurais Artificiais.....	28

III.6.1.1 Redes Neurais atuando como controlador.....	28
III.6.1.2 Controle adaptativo	30
III.6.1.3 Controle preditivo.....	30
III.6.2 Cálculo da ação de controle em um esquema de controle preditivo.....	32
III.7 Esquema de controle utilizado.....	35
CAPÍTULO IV - MATERIAL E MÉTODOS.....	38
IV.1 A planta piloto	38
IV.2 Sistema de controle	40
IV.3 Elaboração do <i>software</i> de redes neurais	42
IV.4 Definição do tamanho da rede.....	43
IV.5 Obtenção dos dados para treinamento e testes.....	43
IV.6 Obtenção dos padrões de treinamento a partir dos dados brutos	44
IV.7 Adimensionalização dos padrões de entrada e saída.....	46
CAPÍTULO V - RESULTADOS E DISCUSSÕES.....	47
V.1 Modelagem dos sistemas usando Redes Neurais Artificiais - treinamento e testes	47
V.1.1 Sistema <i>tanque simples</i>	52
V.1.1.1 Treinamento e testes - sistema <i>tanques simples</i>	52
V.1.1.2 'Treino 1' - <i>Tanque Simples</i>	53
V.1.1.3 Teste referente ao 'treino1' - <i>Tanque Simples</i>	55
V.1.1.4 Comentários.....	56
V.1.1.5 'Treino 2' - <i>Tanque Simples</i>	57
V.1.1.6 Comentários.....	59
V.1.2 Sistema <i>Tanques Acoplados</i>	60
V.1.2.1 Treinamento e testes - <i>Tanques Acoplados</i>	60
V.1.2.2 Treino - <i>Tanques Acoplados</i>	60
V.1.2.3 Teste - <i>Tanques Acoplados</i>	61
V.1.2.4 Comentários.....	64

V.2 Controle	64
V.2.1 Sistema <i>Tanque Simples</i>	64
V.2.1.1 Comentários.....	65
V.2.2 Comportamento da Rede Neural durante o controle do processo	66
V.2.2.1 Comentários.....	66
V.2.3 Sistema <i>Tanques Acoplados</i>	71
V.2.3.1 Comentários.....	72
V.2.4 Comportamento da Rede Neural durante o controle do processo	76
V.2.4.1 Comentários.....	76
CAPÍTULO VI - CONCLUSÕES E SUGESTÕES	78
VI.1 Conclusões.....	78
VI.2 Sugestões para futuros trabalhos	79
APÊNDICE	80
BIBLIOGRAFIA	82

ÍNDICE DE TABELAS

IV.1	Dados genéricos amostrados da planta	44
IV.2	Primeiro padrão de treinamento	45
IV.3	Segundo padrão de treinamento	45
V.1 (a)	Valores finais obtidos para os pesos entre as camadas intermediária e de saída - 'treino1' - <i>tanque simples</i>	53
V.1 (b)	Valores finais obtidos para os pesos entre as camadas de entrada e intermediária - 'treino1' - <i>tanque simples</i>	54
V.2 (a)	Valores finais obtidos para os pesos entre as camadas intermediária e de saída - 'treino2' - <i>tanque simples</i>	57
V.2 (b)	Valores finais obtidos para os pesos entre as camadas de entrada e intermediária - 'treino 2' - <i>tanque simples</i>	58
V.3 (a)	Valores finais obtidos para os pesos entre as camadas intermediária e de saída - treino - <i>tanques acoplados</i>	61
V.3 (b)	Valores finais obtidos para os pesos entre as camadas de entrada e intermediária - treino - <i>tanques acoplados</i>	62

ÍNDICE DE FIGURAS

III.1	Arquitetura de uma <i>Rede Feedforward</i> com 3 camadas.....	13
III.2	Funções de Ativação para Neurônios Artificiais	15
III.3	J-ésimo Neurônio	15
III.4	Fluxograma do algoritmo <i>Backpropagation</i>	22
III.5	Esquema Genérico para Modelagem de processos usando Redes Neurais Artificiais	25
III.6	Esquema para Modelagem de Processos usando ANN's com $m=1$ e $n=2$	26
III.7	Rede Neural Artificial usada como Controlador	28
III.8	Esquema da Arquitetura de Aprendizagem Geral.....	29
III.9	Esquema da Arquitetura de Aprendizagem Especializada	29
III.10	Esquema de Controle Adaptativo	30
III.11	Esquema de Controle Preditivo	31
III.12	Esquema de Controle Proposto.....	32
IV.1	Unidade Experimental	39
IV.2	Diagrama de Blocos do SDCA	41
IV.3	Diagrama de Blocos em Malha Fechada	42
V.1 (a)	Variação na Abertura da Válvula - <i>Tanque Simples</i> - 390 pontos.....	48
V.1 (b)	Variação no Nível de Água - <i>Tanque Simples</i> - 390 pontos.....	48
V.2 (a)	Variação na Abertura da Válvula - <i>Tanque simples</i> - 100 pontos.....	49
V.2 (b)	Variação no Nível de Água - <i>Tanque Simples</i> - 100 pontos.....	49
V.3 (a)	Variação na Abertura da Válvula - <i>Tanques Acoplados</i> - 100 pontos.....	50
V.3 (b)	Variação no Nível de Água - <i>Tanques Acoplados</i> - 100 pontos.....	50
V.4 (a)	Variação na Abertura da Válvula - <i>Tanques Acoplados</i> - 100 pontos.....	51
V.4 (b)	Variação no Nível de Água - <i>Tanques Acoplados</i> - 100 pontos.....	51

V.5	Rede Neural Artificial - 'Treino 1' - <i>Tanque Simples</i>	55
V.6	Rede Neural Artificial - 'Teste 1' Referente ao 'Treino 1' - <i>Tanque Simples</i>	56
V.7	'Treino 2' - <i>Tanque Simples</i>	59
V.8	'Treino' - <i>Tanques Acoplados</i>	63
V.9	'Teste' - <i>Tanques Acoplados</i>	63
V.10 (a)	Mudança no <i>Set Point</i> e Aplicação de Carga - Sistema <i>Tanque Simples</i>	67
V.10 (b)	Ação de Controle - Sistema <i>Tanque Simples</i>	67
V.11 (a)	Mudança no <i>Set Point</i> e Aplicação de Carga - Sistema <i>Tanque Simples</i>	68
V.11 (b)	Ação de Controle - Sistema <i>Tanque Simples</i>	68
V.12 (a)	Mudança no <i>Set Point</i> e Aplicação de Carga - Sistema <i>Tanque Simples</i>	69
V.12 (b)	Ação de Controle - Sistema <i>Tanque Simples</i>	69
V.13	Comparação entre a Saída da Rede e a da Planta - Sistema <i>Tanque Simples</i>	70
V.14	Comparação entre a Saída da Rede e a da Planta - Sistema <i>Tanque Simples</i>	70
V.15	Comparação entre a Saída da Rede e a da Planta - Sistema <i>Tanque Simples</i>	71
V.16 (a)	Mudança no <i>Set Point</i> - Sistema <i>Tanques Acoplados</i>	73
V.16 (b)	Ação de Controle - Sistema <i>Tanques Acoplados</i>	73
V.17 (a)	Mudança no <i>Set Point</i> - Sistema <i>Tanques Acoplados</i>	74
V.17 (b)	Ação de Controle - Sistema <i>Tanques Acoplados</i>	74
V.18 (a)	Mudança no <i>Set Point</i> e Aplicação de Carga - Sistema <i>Tanques Acoplados</i>	75
V.18 (b)	Ação de Controle - Sistema <i>Tanques Acoplados</i>	75
V.19	Comparação entre a Saída da Rede e a da Planta - Sistema <i>Tanques Acoplados</i>	76
V.20	Comparação entre a Saída da Rede e a da Planta - Sistema <i>Tanques Acoplados</i>	77
V.21	Comparação entre a Saída da Rede e a da Planta - Sistema <i>Tanques Acoplados</i>	77

RESUMO

Nos últimos anos técnicas de controle baseadas em estratégias preditivas, adaptativas etc, surgiram como uma opção ao tradicional PID. A maioria destas novas técnicas de controle são dependentes da escolha do modelo que representa o processo, que normalmente é linear ou linearizado. As dificuldades aparecem no controle de sistemas complexos, onde erros de modelagem podem levar ao controle ineficiente, restringindo o uso destes 'novos' controladores. Redes Neurais Artificiais, por sua não linearidade inerente e capacidade de aprender através de exemplos surgem como uma alternativa para resolução destes problemas. Neste trabalho propõe-se a utilização de uma rede neural *Feedforward* Multicamadas em um esquema de controle de processos. O algoritmo de aprendizagem *backpropagation* e a estratégia de otimização *steepest descent* juntamente com uma estratégia de controle preditivo que minimiza uma função custo quadrática foram utilizados neste trabalho como uma estratégia de controle. Testes experimentais realizados em uma unidade piloto onde controlou-se o nível de um tanque simples e de dois tanques acoplados mostraram a eficiência e potencialidade da técnica proposta

SUMMARY

In the last years have appeared, as options to the traditional PID, new control techniques based on predictive and adaptive strategies. Most of these new techniques are depend on the choice of the representative model of the process, which is normally linear or linearized. The difficulties appear on the control of complex system, where modeling mistakes can lead to a non efficient control, limiting the use of these new 'controllers'. Artificial Neural Networks is an alternative to the solution of these problems, for its inherent nonlinearity and its a capacity of learning from examples. In the present work we propose the utilization of a neural network on a process control scheme. The learning back-propagation algorithm associated with the steepest descent method of optimization, with a predictive control strategy, that minimizes a quadratic cost function, is used on this work associated with a control strategy. It is shown the efficiency and the potentiality of the proposed technique, on experimental tests, done in a pilot plant, shown that is possible to control the level of a simple tank and of two jointed tanks.

NOMENCLATURA

$At_{Lp,j}$	Ativação do j-ésimo neurônio da camada L correspondente ao p-ésimo padrão de entrada
At_{Lj}	Ativação do j-ésimo neurônio da camada L
Dif	Valor absoluto da diferença entre $u(k)_{\text{assumido}}$ e $u(k)_{LC}$
E	Erro total entre as saídas da rede e as saídas alvo
E_p	Erro entre a saída da rede e a saída alvo correspondente ao p-ésimo padrão
F	Função não linear qualquer
\hat{F}	Predição para a função F
H	Horizonte de predição
it	it-ésima iteração
n	Ordem do processo em relação a variável controlada
n_{ne}	Número de neurônios na camada de entrada
n_{ni}	Número de neurônios na camada intermediária
n_p	Número de padrões
n_{ns}	Número de neurônios na camada de saída
NU	Horizonte máximo de controle
m	Ordem do processo em relação a variável manipulada
somat(k)	Termo referente a derivada de y_{pred} em relação a $u(k)$
$soma_{Lp,j}$	Somatório das entradas ponderadas do j-ésimo neurônio da camada L correspondente ao p-ésimo padrão de entrada
u	Variável manipulada
$u(k)_{\text{assumido}}$	Valor assumido para u no instante k
$u(k)_{LC}$	Valor de u calculado pela lei de controle no instante k

u_{\max}	Valor máximo da variável manipulada
u_{\min}	Valor mínimo da variável manipulada
$x_{p,i}$	Entrada do i-ésimo neurônio da primeira camada correspondente ao p-ésimo padrão
X_m	Matriz dos padrões de entrada
$w_{(L-1)i,j}$	Conexão entre o i-ésimo neurônio da camada (L-1) e o j-ésimo neurônio da camada L
$\Delta w_{(L-1)i,j}$	Variação da conexão entre o i-ésimo neurônio da camada (L-1) e o j-ésimo neurônio da camada L
y	Variável controlada
y_{\max}	Valor máximo da variável controlada
y_{\min}	Valor mínimo da variável controlada
y_{pred}	Valor de y predito pelo modelo ou pela rede neural artificial
$Y_{p,k}$	Saída alvo para o k-ésimo neurônio da última camada correspondente ao p-ésimo padrão
Y_m	Matriz das saídas alvo
y_{ref}	Trajetória de referência
y_{sp}	<i>Set point</i>

LETRAS GREGAS:

α	Parâmetro de ajuste da trajetória de referência.
λ	Fator de penalização da ação de controle
η	Taxa de aprendizagem do algoritmo <i>backpropagation</i>
θ_{Lj}	Patamar ou limiar de ativação do j-ésimo neurônio da camada L

CAPÍTULO I

INTRODUÇÃO

Nas últimas décadas, técnicas de controle baseadas em modelos preditivos foram bastante estudadas e desenvolvidas. O computador digital teve papel importantíssimo no estreitamento entre a pesquisa de novas técnicas e sua aplicação industrial. Alargaram-se os horizontes do controle de processos de tal forma que a evolução foi inevitável, de técnicas simples, como o clássico PID para outras mais sofisticadas como o LRPC (*Long Range Predictive Control*). Entre as últimas as mais conhecidas são: MAC (*Model Algorithmic Control*), DMC (*Dynamic Matrix Control*) e GPC (*Generalized Predictive Control*), em suas formas simples ou auto ajustáveis, todos já com testes efetivos em plantas industriais.

Todas estas técnicas de controle preditivo são dependentes do modelo, linearizado ou linear. As dificuldades aparecem no controle de sistemas complexos, onde erros de modelagem podem levar ao controle ineficiente do processo. A complexidade, não-linearidade e incertezas são os grandes desafios a serem vencidos. A investigação de outras técnicas de controle que possam vir a contornar estes problemas tornou-se o próximo passo na evolução

As redes neurais ou ANN's (*Artificial Neural Networks*) surgem com a promessa de resolução de alguns problemas comuns encontrados no controle de processos. Sua característica não linear e sua habilidade para aprender através de exemplos as tornam atrativas aos profissionais da área de controle. Nos anos recentes muitos trabalhos têm sido publicados neste

campo. Seu uso na modelagem de processos e também no controle, através de estratégias explícitas ou implícitas, têm sido investigados

A fim de verificar a viabilidade da implementação de ANN's no controle de processos, neste trabalho utilizou-se um modelo de rede neural artificial bastante conhecido, denominado *Feedforward* Multicamadas, juntamente com uma estratégia de controle preditivo que minimiza uma função custo quadrática. Os testes experimentais foram realizados em um processo que, faz parte de um sistema maior compreendido por vários tipos de equipamentos que juntamente com um microcomputador, compõe uma malha de controle em tempo real.

Para uma melhor compreensão dividiu-se a apresentação deste trabalho em capítulos, distribuídos da seguinte forma:

No capítulo II tem-se um breve histórico sobre redes neurais artificiais e sua aplicação a Engenharia Química.

No capítulo III faz-se o detalhamento da rede utilizada, a rede *Feedforward* Multicamadas, e também da teoria sobre a modelagem e controle de processos utilizando ANN's. Neste ponto pode-se dividir este trabalho em duas fases distintas: primeiramente desenvolveu-se um *software* de redes neurais com o objetivo de melhor compreender o funcionamento das mesmas. Também nesta fase realizou-se os treinos e testes com dados retirados de uma planta piloto, mais especificamente de dois sistemas, um tanque simples e dois tanques acoplados. Em uma segunda etapa, utilizou-se os parâmetros obtidos nos treinos para projetar um controlador preditivo e este foi implementado com sucesso para o controle dos sistemas citados.

No capítulo IV apresenta-se a planta e periféricos, bem como a metodologia de tratamento dos dados obtidos para os testes.

No capítulo V discute-se os resultados referentes a modelagem e controle.

Finalmente no capítulo VI são apresentadas as conclusões e perspectivas deste trabalho, onde constata-se a viabilidade do esquema proposto.

CAPÍTULO II

REVISÃO BIBLIOGRÁFICA

Neste capítulo é mostrado um breve histórico sobre redes neurais artificiais desde a década de 40 até os dias atuais, passando por seu emprego na Engenharia Química e destacando alguns trabalhos nesta área.

II.1 - Redes Neurais Artificiais - Uma definição geral

Uma rede neural artificial (ANN- *Artificial Neural Network*) é uma estrutura formada por elementos processadores simples denominados **neurônios** e que são interligados através de canais unidirecionais (pesos) de acordo com a arquitetura escolhida para a rede.

II.2 - Redes Neurais Artificiais - um breve histórico

McCulloch e Pitts (1943), conforme Carvalho Filho *et alli* (1994), propuseram um modelo matemático simplificado de neurônios biológicos. O modelo baseia-se no fato de que em dado instante de tempo, o neurônio ou está disparando ou está inativo, que corresponderia ao zero e um da álgebra booleana. Da maneira proposta o neurônio recebe e produz um valor booleano (0 ou 1). Neste modelo um neurônio j produz um impulso, ou seja, uma saída igual a

1, se e somente se, a soma das entradas for maior que um limiar. A equação II.1 define a função de saída do neurônio de McCulloch-Pitts.

$$At_{Lj} = \left\{ \begin{array}{ll} 1 & \text{se } x = \sum_i w_{(L-1)ij} At_{(L-1)i} - \theta_{Lj} \geq 0 \\ 0 & \text{se } x = \sum_i w_{(L-1)ij} At_{(L-1)i} - \theta_{Lj} < 0 \end{array} \right\} \quad \text{II.1}$$

Onde At_{Lj} e At_{Li} são respectivamente as ativações do neurônio j de uma camada L e do neurônio i da camada $(L-1)$, $w_{(L-1)ij}$ é a conexão entre os neurônios envolvidos e θ_{Lj} é o limiar de ativação do neurônio j .

A partir do modelo proposto por McCulloch-Pitts foram derivados vários outros modelos que permitem a produção de uma saída qualquer, não necessariamente 0 ou 1. (Carvalho Filho *et alli*, 1994).

O interesse no método de aprendizagem foi iniciado por D. O. Hebb, no final da década de quarenta. Ao longo dos anos apareceram formulações matemáticas baseadas nas idéias de Hebb, e que foram chamadas aprendizagens Hebbianas (De Souza Jr., 1993).

Hebb (1949) demonstrou que a capacidade de aprender em redes neurais pode ser conseguida através da variação dos pesos entre os neurônios. A regra de Hebb diz que, quando um estímulo de entrada influencia na produção do estímulo de saída, o peso da conexão entre os neurônios deve ser incrementado. A regra Hebbiana tem sido utilizada em vários algoritmos de aprendizagem, citado por Carvalho Filho *et alli* (1994).

Na década de 60, F. Roseblatt no livro *Principles of Neurodynamics*, forneceu várias idéias a respeito de *Perceptrons*, que são redes de neurônios de limiar, baseados no modelo de McCulloch e Pitts. Também nesta época B. Widrow e M. E. Hoff desenvolveram a Adaline (*ADaptive LINear Element*) com um dispositivo prático para resolver tarefas de reconhecimento

de padrões. O algoritmo de aprendizagem usado é conhecido como *Least Mean Square Error Correction Rule* (LMS), pois minimiza o quadrado dos erros das saídas, (Carvalho Filho *et alli*, 1994).

O *Perceptron* foi bem aceito por sua simplicidade conceitual até que M. Minsk e S. A. Papert, no livro *Perceptrons* (1969) provaram matematicamente que o Perceptron de Roseblatt não podia ser usado para funções lógicas complexas. Os resultados e observações feitas por Minsk e Papert desencorajaram muitas pesquisas na área de redes neurais durante a década de 70 até o início dos anos 80, citado por Clifford (1990).

Na década de 70, Steven Grosseberg na Universidade de Boston e Teuvo Kohonen na Universidade de Helsinki fizeram significativas contribuições. Grosseberg (1976) propôs a Teoria da Resonância Adaptativa (ART - *Adaptive Resonance Theory*). A ART é um sistema que auto organiza padrões de entrada em categorias de reconhecimento, baseado na idéia de que o cérebro espontaneamente se auto organiza dentro de códigos de reconhecimento. Teuvo Kohonen desenvolveu sua idéia sobre mapas auto organizados, baseado na teoria de que os neurônios organizam-se para ajustar-se a vários padrões, citados por Clifford, (1990).

Também na década de 70, Paul Werbos formulou os princípios matemáticos do algoritmo *Backpropagation*, (Clifford, 1990).

David Rumelhart (1986), segundo Cliford (1990), e colaboradores publicaram o livro marco sobre processamento distribuído paralelo, estabelecendo o algoritmo *backpropagation* como o paradigma do campo.

Em grande parte o que motivou as pesquisas na área foi o trabalho publicado por John Hopfield (1982), *Neural Networks and Physical Systems with Emergent Collective Computational Abilities* onde é apresentado um modelo de computação neural baseado na

interação dos neurônios. O modelo consiste de um conjunto de equações diferenciais não lineares de primeira ordem que minimizam uma certa função objetivo, citado por Clifford (1990). Este tipo de rede é agora conhecida como rede de Hopfield.

Existe um grande número de redes sendo estudadas e/ou usadas. Cada qual com suas particularidades e aplicações. Algumas das mais importantes são: *Counterpropagation*, *Adaptive Resonance*, *Neocognitrons*, *Wave Nets*, Rede de *Hopfield* e a mais usada, *Backpropagation*, (Bath et alli, 1990).

Em termos de aplicação de redes neurais para controle de processos, a classe de problemas mais frequentemente abordada tem sido a robótica, e em escala mais modesta, o controle de processos químicos, (De Souza Jr., 1993).

II.3 - Aplicações de Redes Neurais Artificiais na Engenharia Química

Basicamente, a aplicação de redes neurais artificiais nesta área pode ser dividida em três categorias:

- Detecção e diagnóstico de falhas em processos químicos, onde a rede aprende e armazena informações sobre falhas de processo, sendo usada depois para detectar e diagnosticar mau funcionamento de equipamentos e erros de operadores, (De Souza Jr., 1993).

- Modelagem de processos químicos, onde a rede é usada para modelar sistemas não lineares mapeando entradas e saídas conhecidas do processo.

- Controle de processos, onde encontra-se dois tipos de abordagem:

Indireta: onde a rede é utilizada como modelo para prever as saídas do sistema, e a partir daí ajusta-se os parâmetros de um controlador, usando alguma estratégia de controle.

Direta: onde a rede é treinada de forma a atuar diretamente como controlador.

Em Engenharia Química, os primeiros trabalhos publicados foram os de identificação e controle por G. Birky, T. J. McAvoy, N. Bhat, G. Bhagat, e para detecção e diagnóstico de falhas por J. C. Hoskins, D. M. Himmelblau, V. Venkatasubramanian, R. Vaidyanathan, (De Souza Jr., 1993).

Seguem-se alguns dos trabalhos publicados na área de Engenharia Química nos últimos anos.

Venkatasubramanian & Chan (1989) reportaram um estudo no qual uma metodologia baseada em redes neurais foi comparada a uma técnica estatística para detecção de falhas

Watanabe *et alli* (1989) aplicaram redes neurais artificiais para diagnóstico de falhas em um reator químico. A rede usada era composta de dois estágios, um para discriminar as falhas e outro para determinar o grau de severidade das mesmas.

Bath *et alli* (1990) usaram ANN's para modelagem não linear de processos químicos. Três casos foram considerados: um reator em estado estacionário, um tanque agitado com pH dinâmico, ambos simulados, e a interpretação de dados de um biosensor.

Psichogios & Ungar (1991) investigaram o uso de redes neurais artificiais em modelagem e controle de processos. Duas estratégias de controle baseadas em modelos não lineares, IMC (*Internal Model Control*) e MPC (*Model Predictive Control*), foram aplicadas via simulação ao controle de um CSTR exotérmico em um processo SISO (*Single Input-Single Output*).

Lee & Park (1992) apresentaram um esquema de controle *feedforward* utilizando uma rede neural conectada em paralelo a um controlador MPC, cuja saída é resultado da soma da ação de controle para compensar efeitos modelados e da ação de controle para compensar efeitos não modelados. A rede foi treinada *on line* pela minimização da saída do controlador MPC correspondente a efeitos não modelados. O esquema foi aplicado, via simulação, a uma coluna de destilação e ao controle de um reator não linear.

Megan & Cooper (1992) propuseram uma estratégia de controle usando ANN's que utilizava uma abordagem de reconhecimento de padrões em controle adaptativo. Duas redes neurais interconectadas foram treinadas para interpretar padrões de erros resultantes de mudanças no *set point*. Uma primeira rede aprendia a relação entre um dado padrão de erro e o grau de diferença entre o ganho do modelo usado e o ganho atual do processo. Uma segunda rede realizava o mesmo procedimento para a constante de tempo do sistema. As predições da rede foram usadas para ajustar os parâmetros de um controlador PI. A estratégia foi demonstrada via simulação e em testes em escala piloto, usando dois tanques reservatórios em série.

Fan *et alli* (1993) apresentaram uma abordagem para diagnóstico de falhas em processos químicos, usando redes neurais artificiais. A rede proposta foi a *backpropagation*, modificada pela adição de unidades funcionais na camada de entrada, tornando possível o diagnóstico simultâneo de falhas múltiplas e seus correspondentes níveis, bem como ampliando a capacidade da rede aprender relações não lineares complexas. O esquema foi aplicado, via simulação, a um reator em operação no estado estacionário.

Gupta & Narasimhan (1993), fizeram um estudo onde exploraram o uso de redes neurais artificiais para a detecção de erros brutos em medidas de sensores. A performance da estratégia foi comparada a de métodos estatísticos tradicionais.

No Brasil, segundo De Souza Jr (1993), redes neurais artificiais foram usadas por Borges e Castier (1993) para avaliar o desempenho dos modelos termodinâmicos UNIQUAC e UNIFAC no cálculo do equilíbrio líquido-vapor de sistemas binários isotérmicos. Lima Verde *et alli* (1992) usaram ANN's para prever a velocidade de escoamento de sólidos em qualquer ponto de um reator de leito fluidizado circulante.

De Souza Jr (1993) utilizou redes neurais artificiais para prever a área metálica de um catalizador a partir de condições operacionais de manufatura. Neste mesmo trabalho fez a modelagem de um reator não linear isotérmico, no qual ocorria uma reação do tipo A B e de outro de homopolimerização de Vinil Acetato, ambos simulados. As redes treinadas foram usadas em esquemas preditivos para o controle destes sistemas.

Dayal *et alli* (1994) estudaram controladores não lineares baseados em redes neurais artificiais e usando estruturas preditivas do tipo IMC. Os autores implementam três tipos de controladores, um IMC linear, um IMC utilizando uma ANN como modelo direto do processo e outra como modelo inverso para gerar as ações de controle e um terceiro IMC onde utilizaram também uma ANN como modelo direto do processo e o controlador foi obtido pela inversão numérica desta ANN. A implementação, performance e comparação entre estes controladores foram ilustradas em simulações com dois CSTR não lineares.

Chen & Weigand (1994) apresentaram um técnica de otimização dinâmica combinando um modelo de rede neural artificial e uma estratégia UDMC (*Universal Dynamic Matrix Control*). O esquema foi aplicado, via simulação, a dois processos: um reator operando em batelada e um reator bioquímico contínuo.

Watanabe *et alli* (1994) apresentaram um estrutura denominada HANN (*Hierarchical Artificial Neural Network*) que é capaz de dividir um grande número de padrões dentro de poucos subconjuntos podendo assim fazer uma classificação mais eficiente. Com este esquema

foi possível classificar falhas múltiplas e simultâneas em um reator químico, via simulação, a partir do treinamento envolvendo apenas uma falha.

Thompson & Kramer (1994) apresentaram um esquema de modelagem de processos químicos combinando conhecimentos prévios a respeito do processo e redes neurais artificiais. São usados, neste esquema, um modelo simplificado do processo e equações básicas tais como balanço de massa para melhorar as previsões da rede quando os dados de que se dispõe para modelagem são escassos e com ruído. A estratégia foi aplicada, via simulação, ao estudo de um caso de fermentação *fed batch* de penicilina.

Cheng *et alli* (1995) propuseram um método para identificação de processos com tempo morto variável utilizando um rede neural recorrente conhecida como IRN (*Internal Recurrent Neural Network*). A estratégia foi aplicada, via simulação a um processo de neutralização de pH.

Como já comentado, há um grande número de redes neurais artificiais sendo estudadas e usadas, porém, a rede *backpropagation* é sem dúvida a mais empregada nos trabalhos na área de Engenharia Química e também em outras áreas. No desenvolvimento deste trabalho utiliza-se um rede deste tipo, a qual será descrita no próximo capítulo.

CAPÍTULO III

FUNDAMENTOS TEÓRICOS

Na primeira parte deste capítulo procura-se mostrar, de forma objetiva, o que é uma rede neural artificial *feedforward*, sua arquitetura, tipo de neurônio, método de aprendizagem, bem como implementação de ANN's e vantagens e desvantagens de sua utilização.

Na segunda parte enfoca-se a utilização de redes neurais artificiais em controle de processos, onde são mostradas algumas estratégias de modelagem e controle via ANN's.

III.1 - Redes Neurais Artificiais

No capítulo II encontra-se uma definição genérica para redes neurais artificiais a qual pode-se acrescentar que são meios de computação paralela, onde as informações não ficam armazenadas em um ponto específico e sim distribuídas por toda a rede, em suas unidades de processamento local (neurônios artificiais) e suas conexões (pesos).

Em se tratando de redes neurais artificiais as informações podem ser propagadas para frente (redes multicamadas *feedforward*) e também para trás (redes recorrentes), (De souza Jr, 1993). Para este trabalho as redes de interesse são as do primeiro tipo, também conhecidas como *backpropagation*. Esta classe de redes é capaz de aprender através de exemplos, sendo possível

fazer uma aproximação para uma função $y=f(x)$, a partir de dados amostrados de y e x . O termo *backpropagation* provém da regra de aprendizagem usada para o treinamento da rede, que leva o mesmo nome.

III.1.1 - Arquitetura das Redes Neurais Artificiais

A figura III.1 mostra a arquitetura de uma rede neural artificial *feedforward* multicamada. As redes deste tipo são compostas de uma camada de entrada, outra de saída e uma ou mais camadas intermediárias. Cada camada é constituída por um ou mais neurônios artificiais, representados na figura por círculos, as linhas unindo os neurônios são as conexões ou pesos, as setas indicam o sentido do fluxo de informações. A arquitetura da rede mostrada assemelha-se a de algumas das redes de neurônios biológicos. Neste trabalho a rede usada possui apenas uma camada intermediária, pois Hiecht-Nielsen (1989), provou que qualquer função contínua pode ser aproximada para qualquer grau de precisão usando uma rede neural *backpropagation* com três camadas, desde que haja um número suficiente de neurônios ativos na camada escondida, citado por De Souza Jr (1993).

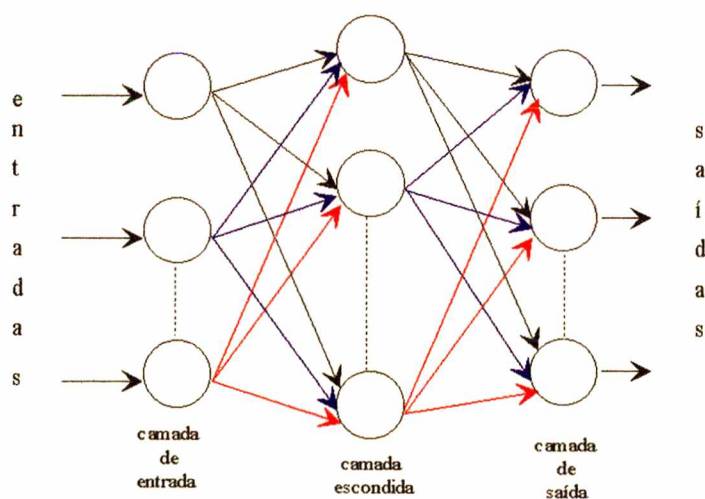


Figura III.1 - Arquitetura de uma rede *feedforward* com 3 camadas

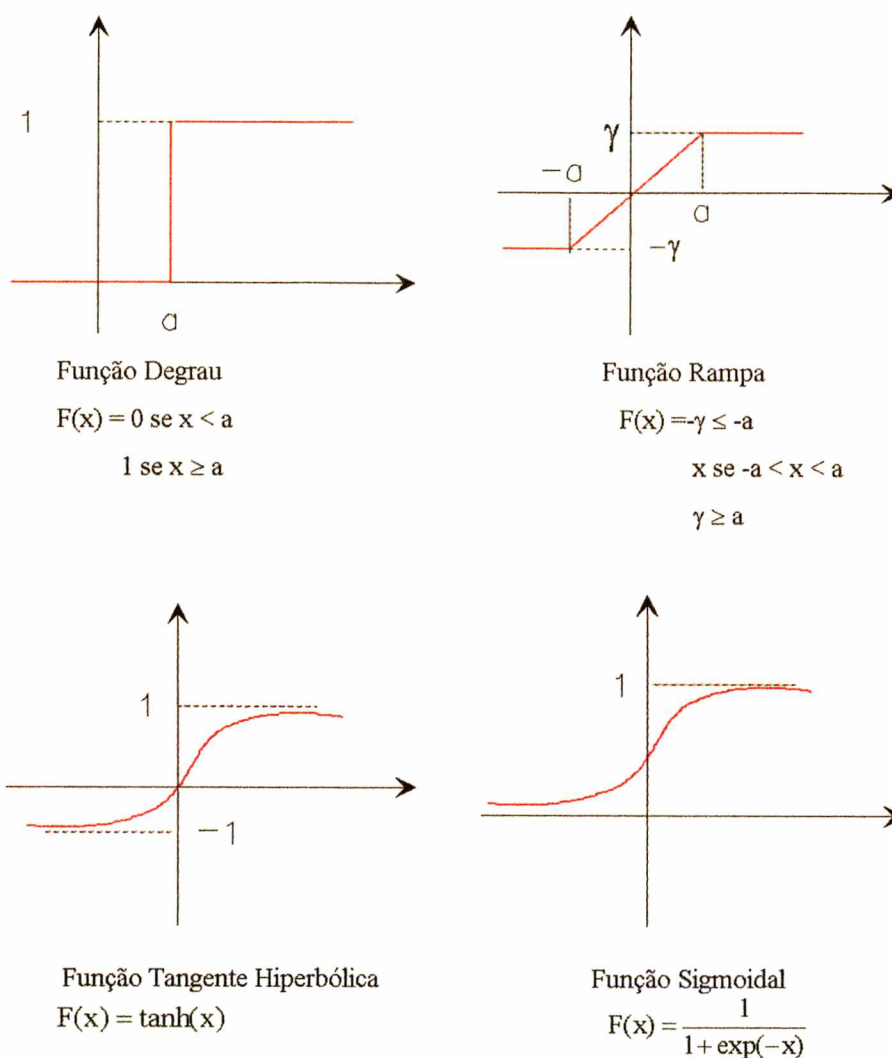
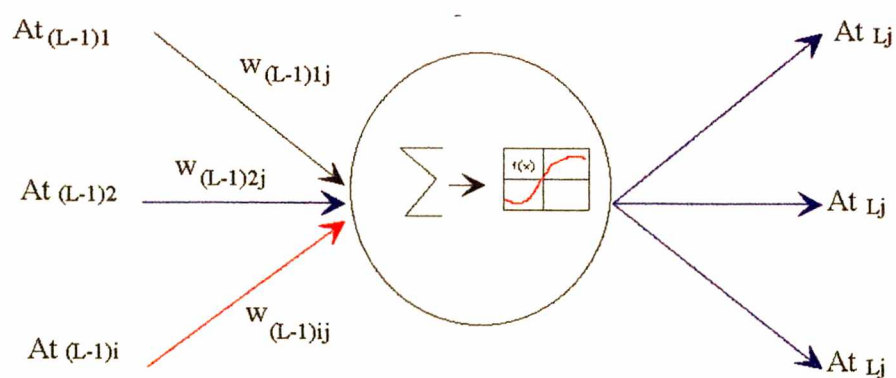


Figura III.2 - Funções de Ativação para Neurônios Artificiais

Figura III.3 - j -ésimo neurônio

Os neurônios da primeira camada recebem e distribuem as entradas, os neurônios da camada intermediária processam as informações vindas da primeira camada. Os neurônios da última camada produzem a saída da rede.

As redes neurais possuem normalmente um neurônio *bias* na camada de entrada e na intermediária. Esses neurônios têm como entrada um valor constante igual a 1 e transmitem este mesmo valor, como ativação, para todos os neurônios da camada seguinte. Os neurônios de uma camada são conectados aos neurônios da camada seguinte através de canais unidirecionais, os pesos.

III.1.2 - Os neurônios artificiais

Os neurônios artificiais são unidades de processamento independentes, cujas respostas às suas respectivas entradas são geradas por funções, chamadas funções de ativação. Os neurônios quando assim representados são ditos **ativos**. Na rede mostrada na figura III.1, apenas os neurônios da segunda e terceira camadas são ativos. Pode-se ter outras combinações de neurônios ativos e inativos para a rede em questão, porém neste trabalho optou-se pelo uso da combinação citada. A figura III.2 mostra algumas das funções de ativação que podem ser usadas para gerar a saída dos neurônios ativos.

Para melhor compreensão, na figura III.3 mostra-se um neurônio j de uma camada L . Este neurônio recebe informações da camada $(L-1)$. Estas informações são as ativações dos neurônios da camada $(L-1)$ ponderadas cada uma por um peso w_{ij} , correspondente à conexão entre os neurônios envolvidos. No neurônio j processam-se as seguintes operações: soma das informações ponderadas que dão entrada neste neurônio e cálculo da ativação ou saída, obtida através da função ativação.

As funções mais utilizadas são as tangente hiperbólica e sigmoideal, mostradas na figura III.2, por serem funções não lineares, diferenciáveis e contínuas, sendo que as duas últimas características são necessárias para a utilização do método de aprendizagem *backpropagation*. Outro ponto importante é que ambas fornecem saídas dentro de um intervalo limitado, sendo assim compatíveis com os sistemas físicos estáveis, (De Souza Jr, 1993). Neste trabalho utilizou-se a função sigmoideal. Matematicamente para um neurônio j da camada L correspondente ao p -ésimo padrão de entrada, ativo, a saída é dada da seguinte forma:

$$At_{Lp,j} = F(soma_{Lp,j}) \quad \text{III.1}$$

onde,

$$soma_{Lp,j} = \sum_i w_{(L-1)i,j} At_{(L-1)p,i} \quad \text{III.2}$$

e

$$F(soma_{Lp,j}) = \frac{1}{1 + \exp(-soma_{Lp,j})} \quad \text{III.3}$$

Estas operações diferem para os neurônios da camada de entrada. Estes neurônios executam a operação de aquisição de dados, sendo atribuída a cada um, uma única entrada. São neurônios inativos. Suas saídas são dadas pela equação III.4, abaixo:

$$At_{1p,i} = x_{p,i} \quad \text{para } i = 1, n_{ne} \quad \text{III.4}$$

onde $x_{p,i}$ é o valor de entrada do neurônio i correspondente ao p -ésimo padrão.

III.2 - O método de aprendizagem

O método *backpropagation*, também conhecido como regra delta generalizada, popularizado por Rumelhart *et alli* (1986), citado por De Souza Jr. (1993), para o treinamento de redes neurais artificiais, é o mais comumente encontrado, principalmente na área de interesse desta dissertação: a resolução de problemas típicos de Engenharia Química e mais especificamente modelagem e controle de processos. Redes que utilizam este método também são usadas em outras aplicações: reconhecimento de padrões visuais, síntese e reconhecimento de fala, análise de sinais de sonar, (Bhat *et alli*, 1990).

III.2.1 - Definição e equações

No algoritmo *backpropagation* têm-se duas etapas: na primeira, os sinais de entrada são propagados para frente a fim de gerar as saídas da rede, enquanto que na segunda, as saídas preditas são comparadas com as saídas reais e o erro é propagado de volta, através da rede. Nesta última etapa acontece o ajuste dos pesos, que é o cerne do método de aprendizagem de redes neurais.

A mudança dos pesos é efetuada na direção que minimiza a seguinte função objetivo:

$$E = \sum_{p=1}^{n_p} E_p \quad \text{III.5}$$

$$E_p = \frac{1}{2} \sum_{k=1}^{n_{ns}} (At_{3p,k} - Y_{p,k})^2 \quad \text{III.6}$$

O índice 3 indica que se trata da saída da última camada, uma vez que a rede usada neste trabalho possui apenas uma camada intermediária.

Deve-se salientar ainda, que o índice p indica **padrão**. Cada padrão representa um conjunto de entradas com suas respectivas saídas desejadas apresentadas à rede. Para cada padrão p ($p=1,2,\dots,n_p$), tem-se um conjunto de entradas $x_{p,i}$ com $i = (1,2,\dots,n_{ne})$ e de saídas desejadas $Y_{p,k}$ com $k = (1,2,\dots,n_{ns})$. As saídas preditas pela rede também são referentes a cada padrão p . As variáveis n_p , n_{ne} , n_{ns} são respectivamente, número de padrões, número de neurônios na camada de entrada e na de saída.

A minimização é feita normalmente usando a estratégia de otimização *steepest descent* (passo descendente), onde a direção de busca é o negativo do gradiente (Himmelblau, 1988). Para o ajuste dos pesos da camada $(L-1)$, que corresponde a conexão entre os neurônios i da camada $(L-1)$ e os neurônios j da camada (L) , para a iteração $(it+1)$, tem-se:

$$w^{it+1}_{(L-1) i,j} = w^{it}_{(L-1) i,j} + \eta \Delta w^{it}_{(L-1) i,j} \quad \text{III.7}$$

onde η é taxa de aprendizagem que pode variar entre 0 e 1, e

$$\Delta w_{(L-1) i,j} = - \frac{\partial E}{\partial w_{(L-1) i,j}} \quad \text{III.8}$$

O ajuste dos pesos entre a camada de saída e a camada intermediária, $w_{2j,k}$, é obtido, usando a regra da cadeia, resolvendo a seguinte derivada, onde E é dado pelo equação III.5:

$$\Delta w_{2j,k} = - \frac{\partial E}{\partial w_{2j,k}} \quad \text{III.9}$$

O índice 2 indica que se trata da conexão entre os neurônios da segunda e terceira camadas

Desta forma, a expressão para a variação dos pesos da camada de saída pode ser dada pela equação abaixo,

$$\Delta w_{2j,k} = \sum_{p=1}^{n_p} (At_{3p,k} - Y_{p,k})(1 - At_{3p,k})At_{2p,j} \quad \text{III.10}$$

e finalmente, o ajuste destes pesos para a iteração (it+1) é efetuado usando a equação seguinte,

$$w_{2j,k}^{it+1} = w_{2j,k}^{it} + \eta \Delta w_{2j,k}^{it} \quad \text{III.11}$$

Da mesma forma, para o ajuste dos pesos entre a camada de entrada e intermediária, $w_{1i,j}$, resolve-se a seguinte derivada, onde E é dada pela equação III.5,

$$\Delta w_{1i,j} = - \frac{\partial E}{\partial w_{1i,j}} \quad \text{III.12}$$

cuja solução é dada pela equação seguinte:

$$\Delta w_{1i,j} = \sum_{p=1}^{n_p} \left[\sum_{k=1}^{n_{ns}} (At_{3p,k} - Y_{p,k}) At_{3p,k} (1 - At_{3p,k}) w_{2j,k} \right] At_{2p,j} (1 - At_{2p,j}) At_{1p,i} \quad \text{III.13}$$

O índice 1 indica que se trata da conexão entre os neurônios da primeira e segunda camadas.

O ajuste dos pesos entre as camadas de entrada e intermediária para a iteração (it+1) é dado pela seguinte equação:

$$w_{1i,j}^{it+1} = w_{1i,j}^{it} + \eta \Delta w_{1i,j}^{it} \quad \text{III.14}$$

III.2.2 - O Algoritmo *backpropagation*

Em algumas situações o algoritmo *backpropagation* baseado no método de gradiente descendente pode não garantir chegar ao erro global mínimo para padrões linearmente dependentes. Contudo, na maioria das vezes, com este método consegue-se atingir o objetivo desejado. Este problema é muito referenciado como problema de erro de mínimo local. O algoritmo *backpropagation* e a arquitetura *feedforward* Multicamadas são, sem dúvida, o paradigma de redes neurais mais utilizado em pesquisas desta área, (Carvalho Filho *et alli*, 1994).

O algoritmo *backpropagation* pode ser enunciado como segue:

Passo 1 - Apresente os valores dos pesos

Apresente os pesos aleatórios distribuídos entre -1 e 1, se for a primeira iteração, caso contrário apresente os pesos corrigidos. Ative o **passo 2**

Passo 2 - Apresente um Padrão de Entrada e a Saída Desejada

Utilizando uma determinada estratégia apresente um padrão de entrada e sua respectiva saída. Ative o passo **passo 3**

Passo 3 - Calcule a Saída

Permita que cada camada produza os valores de saída até que a última das camadas seja atingida e o padrão de saída predito pela rede, seja obtido. Calcule o erro entre o valor predito pela rede e o valor real. Se o padrão apresentado for o último, ative o **passo 4**, caso contrário ative o passo **passo 2**

Passo 4 -*Cheque Magnitude do Erro Total*

Para checar a magnitude do erro, pode-se adotar várias estratégias diferenciadas quanto a forma de aceitar o erro como desprezível. Se a condição é satisfeita, então a rede aprendeu o conjunto de treinamento, caso contrário ative o *passo 5*.

Passo 5 - *Ajuste Pesos $w_{2j,k}$*

Atualize os pesos entre as camadas intermediária e de saída, utilizando as equações III.10 e III.11. Ative o *passo 6*

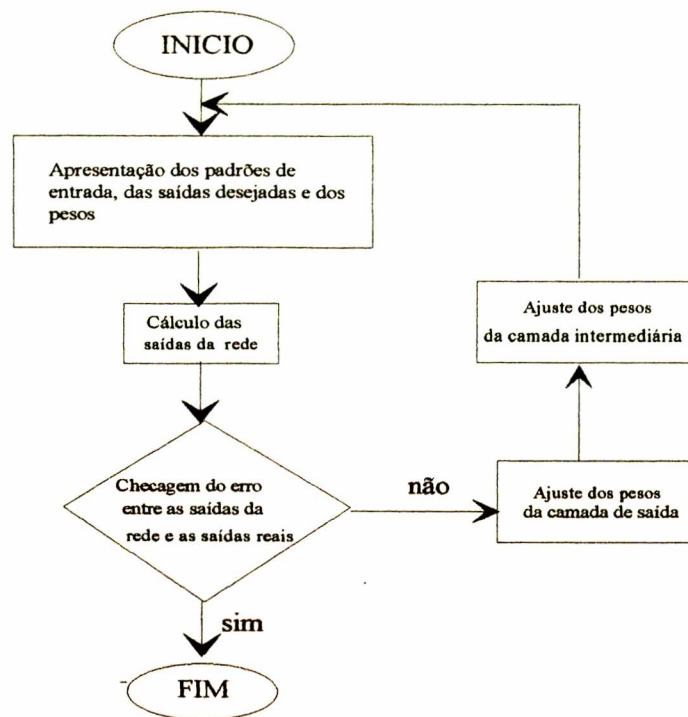
Passo 6 - *Ajuste Pesos $w_{1i,j}$*

Atualize os pesos entre a camada de entrada e a de saída, utilizando as equações III.13 e III.14. Ative o *passo 1*

O fluxograma do algoritmo *backpropagation* é mostrado na figura III.4.

III.2.3 - Utilização da rede após o treinamento

Terminada a fase de treino, a rede é usada em testes para verificar a confiabilidade dos pesos obtidos. Nesta fase, apenas os passos de 1 a 3, mostrados anteriormente, são efetivamente executados. Os pesos usados são os calculados durante o treinamento. Uma vez testada e aprovada, a rede atua na predição da saída para valores de entrada para os quais não foi treinada e cuja saída real é desconhecida.

Figura III.4 - Fluxograma do algoritmo *backpropagation*

III.3 - Implementação de Redes Neurais

Redes neurais podem ser implementadas em *software*, *hardware*, meios óticos ou também em sistemas híbridos combinados. A escolha dependerá das necessidades associadas ao projeto de implementação, bem como da disponibilidade de recursos para ser investido no projeto de implementação, tendo em vista a ponderação entre flexibilidade, desempenho e custo, (Carvalho Filho *et alli*, 1994).

O Caminho natural é partir de uma implementação em *software*. Porém, a tarefa de escrever um *software* para realizar experimentos com redes neurais, representa uma parte significativa na parcela de tempo gasto nas pesquisas nesta área, (Carvalho Filho *et alli*, 1994).

As principais rotinas de implementação de um modelo de redes neurais em *software* podem ser resumidas da seguinte forma, segundo Carvalho filho *et alli* (1994):

1. Uma ***Rotina Principal*** que recebe os parâmetros a serem utilizados na simulação, tais como: número de camadas, número de neurônios por camada, conectividade, tipo de neurônio, tamanho do conjunto de treinamento, dimensão do padrão de entrada, etc. Este procedimento também controla o fluxo de processamento de outras rotinas.
2. Uma ***Rotina de Construção de Rede*** que cria a arquitetura da rede.
3. Uma ***Rotina de Conexão*** que conecta todos os neurônios.
4. Uma ***Rotina de Alocação de Memória*** que reserva as células de memória necessárias para as estruturas de dados da rede.
5. Uma ***Rotina de Aprendizagem*** que mostra à rede os padrões de treinamento, segundo uma dada ordem sequencial. Quando necessário, mostra as saídas desejadas.
6. Uma ***Rotina de Uso*** que apresenta os padrões de teste e obtém a resposta da rede.
7. Uma ***Biblioteca de Rotinas*** que possui a especificação do tipo de dados, rotinas de entrada e saída e outros procedimentos.

III.4 - Vantagens e desvantagens da utilização de Redes Neurais Artificiais.

Algumas das principais vantagens são (Carvalho filho *et alli*, 1994):

1 - Aprendizagem através de exemplos. Um sistema de computação que utiliza uma aprendizagem por exemplos é desejado, quando a complexibilidade computacional do problema a ser resolvido é alta, no sentido de que o problema não possui todas as suas variáveis conhecidas.

2 - Independência do modelo. Uma vez que uma rede neural antes da aprendizagem não possui nenhum conhecimento sobre o problema que se pretende resolver, a mesma rede pode ser aplicada a problemas diferentes sem necessidade de qualquer modificação na sua estrutura básica.

3 - Obtenção dos resultados desejados. Caso uma rede neural não esteja fornecendo resultados aceitáveis, a sua arquitetura pode ser modificada em busca de otimizações.

4 - Processamento distribuído e paralelo. A própria arquitetura da rede e a natureza dos neurônios propiciam um processamento distribuído, paralelo e local. O paralelismo se dá a níveis de módulos de redes neurais, camadas, neurônios e conexões.

Por outro lado pode-se citar como desvantagens:

1 - Falta e dificuldade de um formalismo na especificação e na análise de modelos de redes neurais. Para se compreender os mecanismos fundamentais das redes é necessário realizar simulações que, na maioria das vezes, são tarefas árduas e distantes da realidade do modelo.

2 - Problema de mínimos locais. A utilização de métodos de treinamento baseados no cálculo do gradiente, como o *backpropagation*, pode conduzir o valor da função objetivo a um mínimo local, de onde torna-se difícil escalar a superfície para buscar a verdadeira solução.

3 - **Extrapolações.** Nem sempre a rede consegue bons resultados para valores para os quais não foi treinada, sendo necessário o uso de um conjunto de dados bastante significativo para seu treinamento.

III.5 - Modelagem de processos utilizando Redes Neurais.

A figura III.5 mostra um esquema de modelagem de processos utilizando redes neurais artificiais. Supõe-se neste caso que o processo não linear possa ser descrito pela seguinte equação de diferenças, onde u é a variável manipulada, y , a variável controlada e F uma função não linear qualquer:

$$y(k+1) = F[y(k), y(k-1), \dots, y(k-n+1), u(k), u(k-1), \dots, u(k-m+1)] \quad \text{III.15}$$

Dessa forma, a saída da rede será descrita como:

$$y_{\text{pred}}(k+1) = \hat{F}[y(k), y(k-1), \dots, y(k-n+1), u(k), u(k-1), \dots, u(k-m+1)] \quad \text{III.16}$$

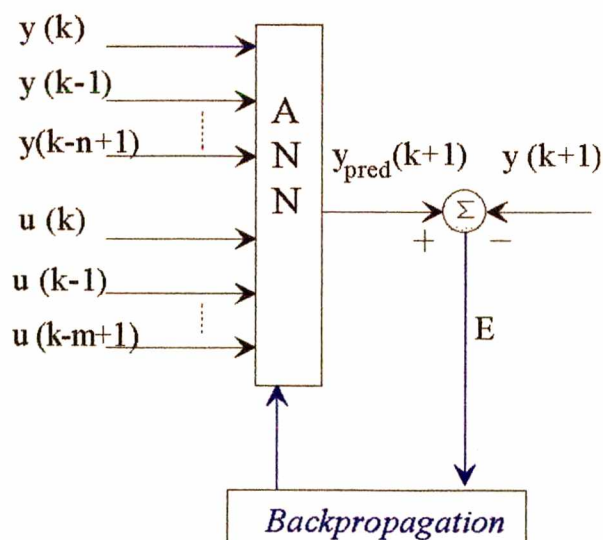


Figura III.5 - Esquema genérico para modelagem de processos usando ANN's

Os valores de $y(k)$, $y(k-1), \dots, y(k-n+1)$, $u(k)$, $\dots, u(k-1), \dots, u(k-m+1)$ correspondem ao padrão de entrada e os de $y(k+1)$ padrão de saída apresentado à rede como alvo. O erro usado para o treinamento da rede, E , é dado pela equação III.5. O erro para cada padrão p pode ser escrito da seguinte forma:

$$E_p = \frac{1}{2} [y_{\text{pred}}(k+1) - y(k+1)]^2 \quad \text{III.17}$$

O número de neurônios na camada de entrada é definido pela escolha de m e n . Para este trabalho os sistemas foram representados pela equação III.15 com $m=1$ e $n=2$. Dessa forma a equação III.16, ou seja a saída da rede será representada por:

$$y_{\text{pred}}(k+1) = \hat{F}[y(k), y(k-1), u(k)] \quad \text{III.18}$$

Esta equação foi escolhida por ser bastante genérica e não possuir nenhum termo linearizado. Os valores de m e n foram escolhidos considerando que a rede neural projetada para modelar os sistemas seria capaz de generalizar o comportamento dos mesmos utilizando um número pequeno de variáveis de entrada. O esquema mostrado na figura III.5 pode ser redefinido como segue:

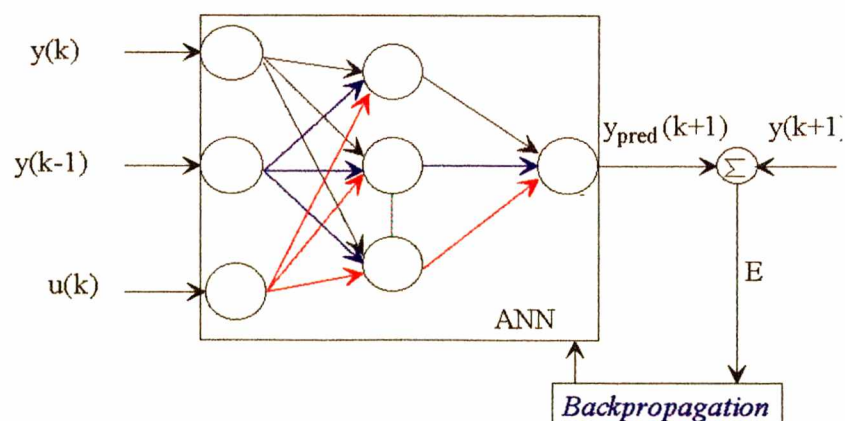


Figura III.6- Esquema para modelagem de processos usando ANN's com $m=1$ e $n=2$

O treinamento usando este tipo de apresentação pode ser feito *on line* ou *off line*

a) **Treinamento *off line*:** os dados são previamente obtidos da planta e a partir daí define-se um número finito de padrões para o treinamento, os quais são apresentados à rede e o ajuste dos pesos é efetuado usando o método *backpropagation*.

b) **Treinamento *on line*:** os padrões são apresentados em tempo real. A cada intervalo de amostragem um novo dado é acrescentado e também faz-se o ajuste dos pesos.

Neste trabalho optou-se pelo treinamento *off line*.

III.6 - Controle de processos baseado em Redes Neurais Artificiais

A evolução na área de controle tem sido motivada por três grandes necessidades: a necessidade de tratar sistemas complexos, de acompanhar o aumento nas exigências de projeto e de atender essas exigências utilizando o menor número de informações a respeito da planta e seus arredores, (Antsaklis, 1990).

O aumento da complexidade dos sistemas dinâmicos e a necessidade de controlá-los sob condições incertas traz a tona o questionamento dos métodos convencionais de controle. Sob este prisma, novas técnicas tem sido investigadas para melhorar a performance de controladores tradicionais. Uma grande área de estudos é a dos controladores adaptativos onde pode-se incluir o STC (*Self Tuning Controller*), Clarke & Gawthrop (1979) e o GPC (*Generalized Predictive Control*), Clarke *et alli* (1987), em sua forma auto ajustável.

As Redes Neurais Artificiais, por sua inerente não linearidade surgem para oferecer novas direções para melhor entender e talvez resolver problemas de controle até então considerados difíceis, (Antsaklis, 1990). A atratividade das ANN's reside no fato de que uma

vez que se possa obter dados de entrada e saída do sistema, uma rede neural pode ser treinada para modelar o processo ou mesmo para atuar diretamente como controlador

III.6.1 - Estratégias de controle usando Redes Neurais Artificiais

Basicamente pode-se dividir as estratégias de controle usando redes neurais em duas classes. Uma na qual a rede é usada como controlador e outra onde atua como modelo interno do processo, a partir do qual projeta-se o controlador.

III.6.1.1 - Rede Neural atuando como controlador

A rede é usada diretamente como controlador. O problema deste esquema, mostrado na figura III.7, é realizar o treinamento da rede. Deve-se lembrar que o erro utilizado para o ajuste dos pesos no método *backpropagation* é a diferença entre a saída da rede e um valor alvo, que deve ser conhecido a priori. Neste caso, em que a rede é usada como controlador, esse alvo é a ação de controle que levará a planta ao estado desejado. Tal valor é desconhecido. O único erro do qual se dispõe é o erro entre a saída da planta e o *set point*.

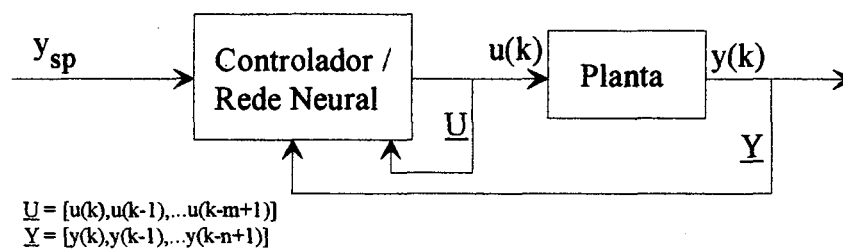


Figura III.7 - Rede Neural Artificial usada como Controlador

De Souza Jr. (1993), mostra algumas maneiras de treinar a rede que será usada como controlador. Entre elas pode-se destacar a "arquitetura de aprendizagem geral", mostrada na figura III.8:

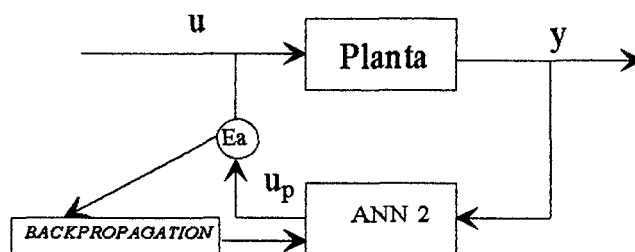


Figura III.8 - Esquema da Arquitetura de Aprendizagem Geral

Neste tipo de arquitetura a rede é treinada para obter a inversa do sistema. Saídas conhecidas da planta são alimentadas à rede que gera uma saída u_p . Esta saída é comparada com o valor da ação de controle u que levou a planta ao estado y . O treino da rede é feito *off line*, retropropagando o erro E_a . A rede treinada é usada como controlador.

Um outro arranjo é mostrado na figura III.9 onde a planta seria simulada por uma rede neural previamente treinada, como proposto por Nguyen (1990). O arranjo formado pelas duas redes pode ser compreendido como um único bloco, e sendo assim o erro entre a saída da planta e o *set point* pode ser usado para ajustar os pesos da rede/controlador, (ANN 1), uma vez que esse erro pode ser retropropagado através da rede que simula a planta, (ANN 2).

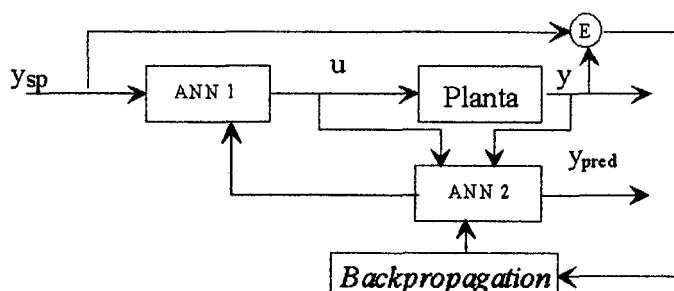


Figura III.9 - Esquema da Arquitetura de Aprendizagem Especializada

III.6.1.2 - Controle adaptativo

Neste esquema, uma rede neural artificial é usada como modelo interno para o projeto de um controlador auto ajustável. A cada instante de amostragem a rede procede no cálculo de y_{pred} , como mostra a equação abaixo:

$$y_{pred}(k) = \hat{F}[y(k-1), y(k-2), \dots, y(k-n+1), u(k-1), u(k-2), \dots, u(k-m+1)] \quad \text{III.19}$$

A cada período de amostragem os pesos também são ajustados para minimizar o erro entre a saída da rede e a saída da planta. Os novos pesos são utilizados para modificar os parâmetros do controlador. O esquema é mostrado na figura III.10. Um outro esquema de controle adaptativo pode ser visto em Chen (1990).

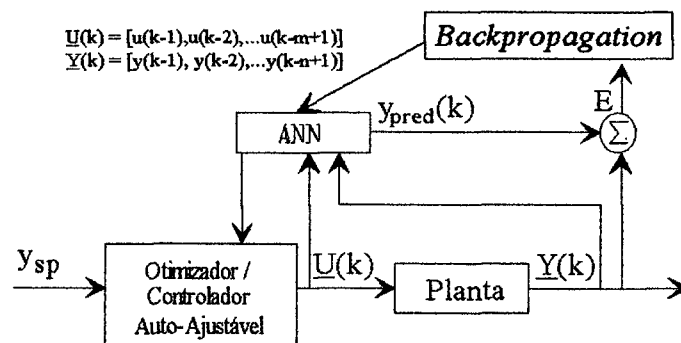
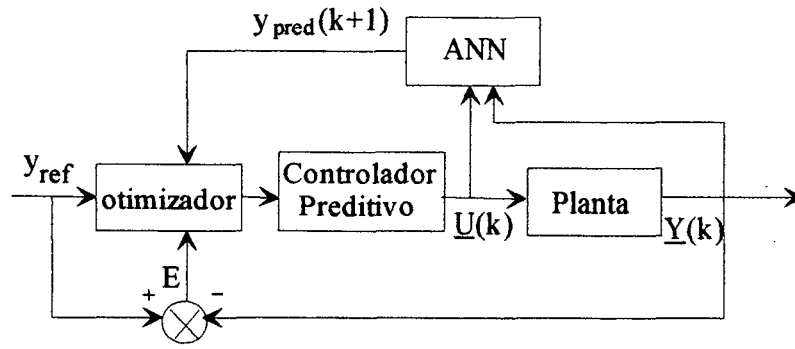


Figura III.10- Esquema de Controle Adaptativo

III.6.1.3 - Controle preditivo

As ANN's são utilizadas em controle preditivo da mesma forma que em controle adaptativo: como modelo do processo, no entanto, os pesos não são reajustados a cada instante de amostragem. Neste esquema, mostrado na figura III.11, a rede prediz as saídas futuras da planta com o treinamento feito *off line* como descrito no item III.5.



onde:

$$\underline{U}(k) = [u(k), u(k-1), \dots, u(k-m+1)]$$

$$\underline{Y}(k) = [y(k), y(k-1), \dots, y(k-n+1)]$$

Figura III.11 - Esquema de Controle Preditivo

A abordagem é essencialmente a mesma usada para o projeto de um controlador DMC (Dynamic Matrix Control), exceto que um modelo neural não-linear é usado no lugar de um modelo de convolução linear. Basicamente o otimizador calcula as ações de controle futuras para minimizar uma função objetivo (Bath *et alli*, 1990). Como exemplo pode ser citada a função abaixo:

$$J(k) = \sum_{i=1}^H [y_{\text{pred}}(k+i) - y_{\text{ref}}(k+i)]^2 + \lambda \sum_{j=1}^{NU} [\Delta u(k+j-1)]^2 \quad \text{III.20}$$

onde é empregada a minimização de um critério quadrático de custo para calcular o vetor de ações de controle que conduza a trajetória do modelo à trajetória de referência. Para o caso deste trabalho a trajetória de referência é dada por:

$$y_{\text{ref}}(k) = y(k) \quad \text{III.21}$$

$$y_{\text{ref}}(k+i) = \alpha y_{\text{ref}}(k+i-1) + (1-\alpha)y_{\text{sp}} \quad i=1, \dots, H \quad \text{III.22}$$

Na equação III.20, λ é um fator de penalização que reduz as variações nos incrementos da variável manipulada, H é o horizonte de predição e NU é o horizonte de controle. Na equação III.22, α é um parâmetro de ajuste que varia de 0 a 1.

III.6.2 - Cálculo da ação de controle em um esquema de controle preditivo

Neste trabalho utilizou-se a função custo mostrada na equação III.20 com $H=1$ e $NU=1$, a trajetória de referência foi dada pelas equações III.21 e 22. A ação de controle é calculada minimizando a equação III.20 em relação a $u(k)$, como mostra a equação III.23, e para maior simplicidade, escreve-se $y_{\text{pred}}(k+1) = y_{\text{pred}}$, uma vez que a rede prevê apenas 1 (um) passo no futuro e possui apenas 1(um) neurônio na camada de saída.:

$$\frac{\partial J(k)}{\partial u(k)} = 2[y_{\text{ref}}(k+1) - y_{\text{pred}}] \frac{\partial y_{\text{pred}}}{\partial u(k)} + 2\lambda[u(k) - u(k-1)] \quad \text{III.23}$$

para obter-se a derivada de y_{pred} em relação a $u(k)$, é preciso lembrar que y_{pred} é o valor de saída da rede e é dado por,

$$y_{\text{pred}} = \frac{1}{1 + \exp[-(\text{soma}_{3p,1})]} \quad \text{III.24}$$

onde,

$$\text{soma}_{3p,1} = \sum_{j=1}^{n_{ni}} w_{2j,1} At_{2p,j} \quad \text{III.25}$$

assim, utilizando a regra da cadeia, pode-se escrever,

$$\frac{\partial y_{\text{pred}}}{\partial u(k)} = \frac{\partial y_{\text{pred}}}{\partial \text{soma}_{3p,1}} \frac{\partial \text{soma}_{3p,1}}{\partial u(k)} \quad \text{III.26}$$

decompondo a equação anterior tem-se,

$$\frac{\partial y_{\text{pred}}}{\partial \text{soma}_{3p,1}} = \frac{\exp(-\text{soma}_{3p,1})}{[1 + \exp(-\text{soma}_{3p,1})]^2} \quad \text{III.27}$$

$$\frac{\partial \text{soma}_{3p,1}}{\partial u(k)} = \sum_{j=1}^{n_{ni}} w_{2j,1} \frac{\partial \text{At}_{2p,j}}{\partial u(k)} \quad \text{III.28}$$

a equação III.27 pode ser reescrita, após algum algebrismo, como mostrado abaixo:

$$\frac{\partial y_{\text{pred}}}{\partial \text{soma}_{3p,1}} = y_{\text{pred}} (1 - y_{\text{pred}}) \quad \text{III.29}$$

utilizando as equações III.28 e III.29 podemos reescrever a equação III.26 conforme mostrado abaixo,

$$\frac{\partial y_{\text{pred}}}{\partial u(k)} = y_{\text{pred}} (1 - y_{\text{pred}}) \sum_{j=1}^{n_{ni}} w_{2j,1} \frac{\partial \text{At}_{2p,j}}{\partial u(k)} \quad \text{III.30}$$

Para resolver a equação III.30 é preciso calcular a derivada de $\text{At}_{2p,j}$ em relação a $u(k)$, para tanto devemos lembrar que a saída de um neurônio j da camada intermediária é dada conforme a equação seguinte,

$$\text{At}_{2p,j} = \frac{1}{1 + \exp(-\text{soma}_{2p,j})} \quad \text{III.31}$$

onde,

$$\text{soma}_{2p,j} = \sum_{i=1}^{n_{ne}} w_{li,j} At_{lp,i} \quad \text{III.32}$$

assim, usando a regra da cadeia, obtem-se

$$\frac{\partial At_{2p,j}}{\partial u(k)} = \frac{\partial At_{2p,j}}{\partial \text{soma}_{2p,j}} \frac{\partial \text{soma}_{2p,j}}{\partial u(k)} \quad \text{III.33}$$

resolvendo-se separadamente os membros do lado direito da equação III.33, tem-se

$$\frac{\partial At_{2p,j}}{\partial \text{soma}_{2p,j}} = At_{2p,j}(1 - At_{2p,j}) \quad \text{III.34}$$

$$\frac{\partial \text{soma}_{2p,j}}{\partial u(k)} = \sum_{i=1}^{n_{ne}} w_{li,j} \frac{\partial At_{lp,i}}{\partial u(k)} \quad \text{III.35}$$

Uma vez que entrada $u(k)$ é apresentada somente ao terceiro neurônio da primeira camada, a equação anterior torna-se:

$$\frac{\partial \text{soma}_{2p,j}}{\partial u(k)} = \sum_{i=1}^{n_{ne}} w_{li,j} \frac{\partial At_{lp,i}}{\partial u(k)} \quad \text{III.36}$$

acho que o somatório n existe

Substituindo-se as equações III.34 e III.36 na equação III.30 obtem-se:

$$\frac{\partial y_{pred}}{\partial u(k)} = y'_{pred} \sum_{j=1}^{n_{ni}} w_{2j,1} At_{2p,j}(1 - At_{2p,j}) w_{13,j} \frac{\partial At_{lp,3}}{\partial u(k)} \quad \text{III.37}$$

dessa forma, a expressão para a derivada da função custo em relação a $u(k)$ pode ser representada pela equação seguinte:

$$\frac{\partial J(k)}{\partial u(k)} = -2[y_{\text{ref}}(k+1) - y_{\text{pred}}] y'_{\text{pred}} \sum_{j=1}^{n_{\text{ni}}} w_{2,j,1} At'_{2p,j} w_{13,j} \frac{\partial At_{1p,3}}{\partial u(k)} + 2\lambda[u(k) - u(k-1)] \quad \text{III.38}$$

onde,

$$y'_{\text{pred}} = y_{\text{pred}} (1 - y_{\text{pred}}) \quad \text{III.39}$$

$$At'_{2p,j} = At_{2p,j} (1 - At_{2p,j}) \quad \text{III.40}$$

$$\frac{\partial At_{1p,3}}{\partial u(k)} = 1 \quad \text{III.41}$$

Assim, igualando-se a equação III.38 a zero, chega-se a expressão para a lei de controle, que é dada pela equação abaixo:

$$u(k)_{\text{LC}} = u(k-1) + \frac{1}{\lambda} [y_{\text{ref}}(k+1) - y_{\text{pred}}] y_{\text{pred}} (1 - y_{\text{pred}}) \sum_{j=1}^{n_{\text{ni}}} w_{2,j,1} At_{2p,j} (1 - At_{2p,j}) w_{13,j} \quad \text{III.42}$$

III.7 - Esquema de controle utilizado

A estratégia adotada é do tipo explícita, uma vez que primeiramente faz-se a predição da saída do sistema através do modelo e depois calcula-se a ação de controle com base nesta saída e nos parâmetros do modelo. A equação III.42 pode ser reescrita substituindo y_{ref} pela equação III.22, com $H=1$, que descreve a trajetória de referência.

$$u(k)_{\text{LC}} = u(k-1) + \frac{1}{\lambda} [\alpha y(k) - (1-\alpha)y_{\text{sp}} - y_{\text{pred}}] \text{somat}(k) \quad \text{III.43}$$

onde,

$$\text{somat}(k) = y_{\text{pred}} (1 - y_{\text{pred}}) \sum_{j=1}^{n_{ni}} w_{2j,1} At_{2p,j} (1 - At_{2p,j}) w_{13,j} \quad \text{III.44}$$

O termo $\text{somat}(k)$, equação III.31, engloba os pesos da rede, bem como sua predição para a saída da planta, y_{pred} , e as saídas dos neurônios da camada intermediária, $At_{2p,j}$. Este termo provém da derivada de y_{pred} em relação $u(k)$. Observa-se que este termo é dinâmico, uma vez que as saídas da rede e dos neurônios intermediários são calculadas a cada instante de amostragem. Apenas os pesos são fixos.

Observa-se que na equação III.30, para calcular $u(k)$ é necessário obter-se y_{pred} ou $y_{\text{pred}}(k+1)$ no instante k . Para tanto é preciso fornecer à rede os valores de $y(k)$, $y(k-1)$ e $u(k)$, sendo que este último é desconhecido, pois é a ação de controle que deverá ser aplicada à planta no instante k . Para contornar este problema pode-se tentar explicitar $u(k)$ a partir de y_{pred} . Esta tarefa torna-se difícil uma vez que a rede é um elemento altamente não linear, cujas saídas da maioria dos neurônios é dada pela função sigmoidal.

Optou-se pela implementação de um esquema iterativo. Para isso, assume-se um valor para u no instante k , e esta entrada juntamente com as variáveis conhecidas, $y(k)$ e $y(k-1)$ são apresentadas à rede que gera uma saída y_{pred} . Esta saída pode estar longe da saída real que o sistema terá caso implemente-se $u(k)$ assumido. Então recalcula-se $u(k)$ através da lei de controle, equação III.29 ou III.30, pois agora todos os termos são conhecidos. Em seguida, faz-se o cálculo da diferença entre $u(k)$ assumido e $u(k)_{LC}$:

$$\text{Dif} = \text{Abs} [u(k)_{\text{assumido}} - u(k)_{LC}] \quad \text{III.45}$$

Se a diferença for considerada suficientemente pequena, ou seja, alcançar a precisão desejada, assume-se que y_{pred} pela rede é igual a $y(k+1)$ do sistema. Assim implementa-se $u(k)_{LC}$. O esquema é mostrado na figura III.12

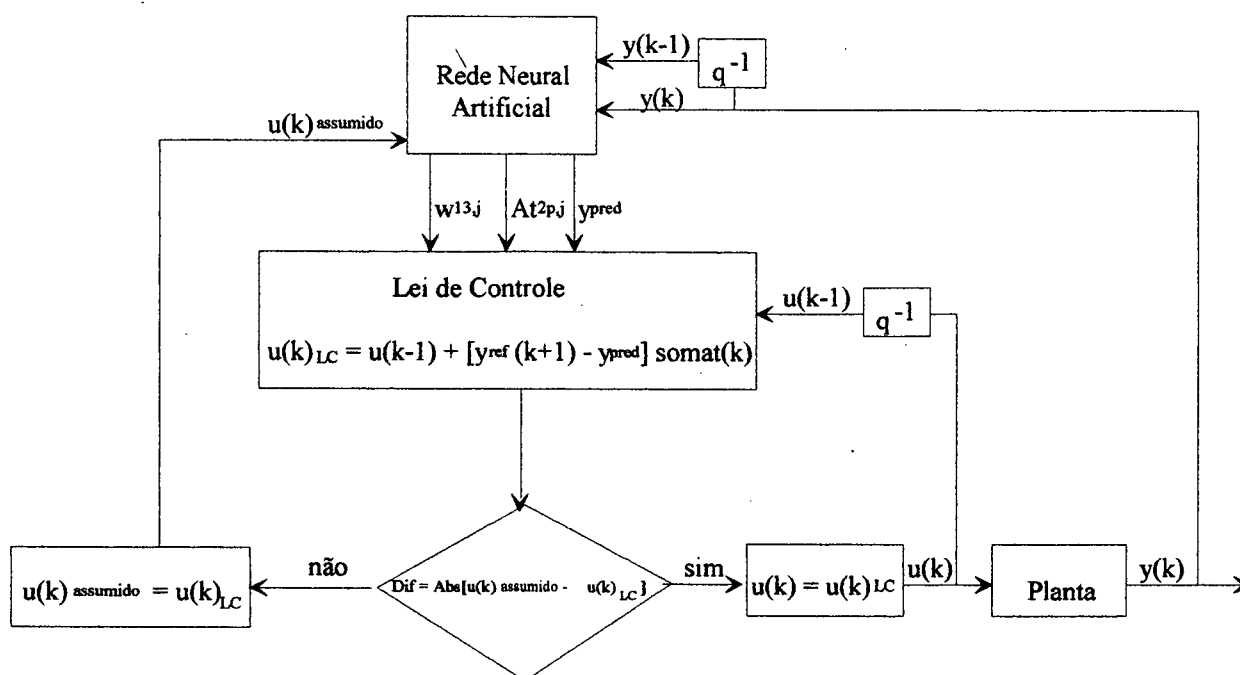


Figura III.12 - Esquema de Controle Proposto

O esquema de modelagem usando redes neurais artificiais, bem como o esquema de controle preditivo descrito neste capítulo foi implementado, neste trabalho, para o controle de nível de dois sistemas, um tanque simples e dois tanques acoplados.

CAPÍTULO IV

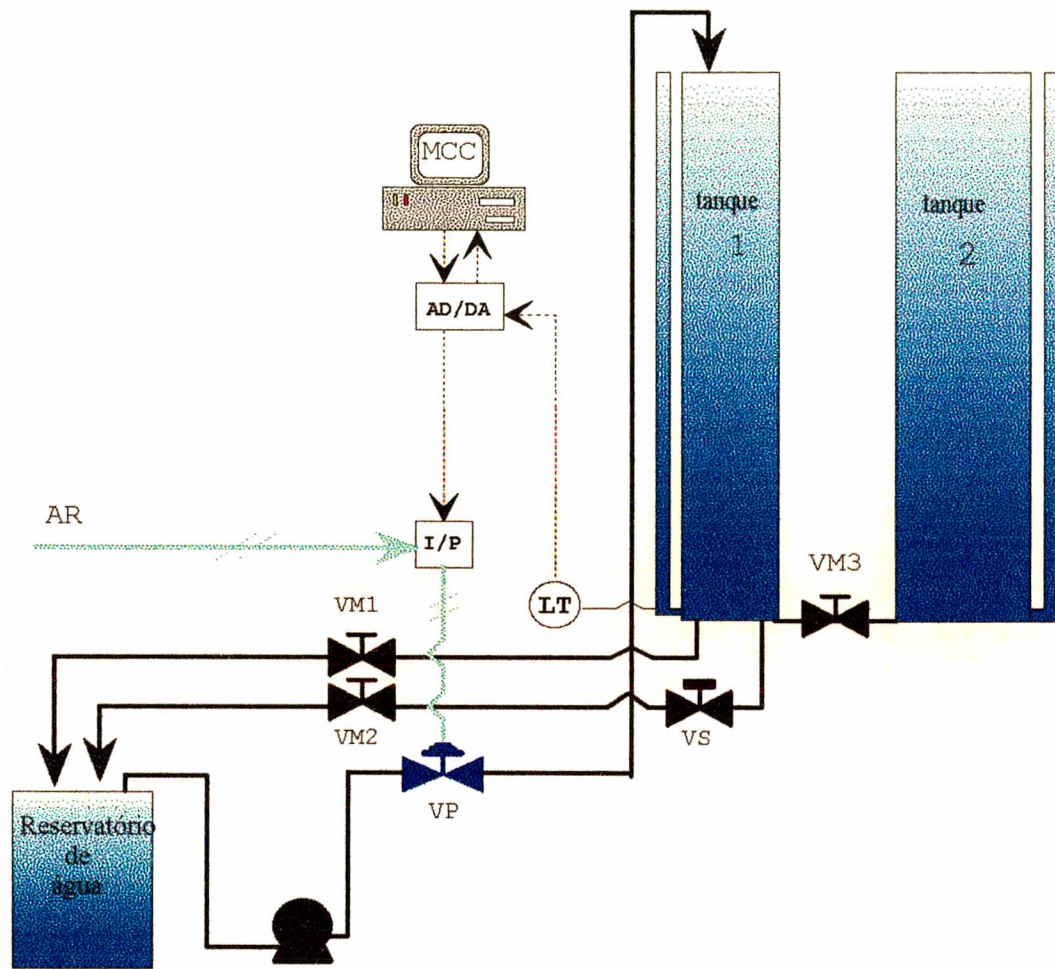
MATERIAL E MÉTODOS

Neste capítulo é mostrado, primeiramente, um esquema do equipamento utilizado no desenvolvimento do trabalho. Também é apresentada a metodologia usada para a aquisição e tratamento dos dados, treinamento e testes da rede neural e controle dos processos.

IV.1 - A Planta piloto

A unidade experimental, mostrada na figura IV.1 é composta por dois tanques de PVC, tendo ambos 2,00 metros de altura. O tanques 1 e 2 possuem diâmetro interno de 0,10 e 0,15 m, respectivamente. O conjunto é alimentado com água da rede hidráulica do laboratório. Possui uma válvula pneumática na linha de alimentação e uma válvula solenóide de acionamento manual numa das linhas de saída. Ambos os tanques possuem mostrador que permite a visualização do nível de líquido em seu interior.

A unidade descrita está instalada no Laboratório de Controle de Processos do Departamento de Engenharia Química da Universidade Federal de Santa Catarina, sendo utilizada para desenvolvimento e aplicação de estratégias modernas de controle de processos e no ensino de controle em tempo real para o curso de Engenharia Química da UFSC.



- MCC = Microcomputador/controlador
 AD/DA = Placa Analógico/Digital-Digital/Analógico
 I/P = Conversor Eletro-Pneumático
 LT = Medidor de Nível
 VM1 = Válvula Manual
 VM2 = Válvula Manual
 VM3 = Válvula Manual
 VS = Válvula Solenóide
 VP = Válvula Pneumática
 Rede Pneumática
 ————— Rede Hidráulica
 - - - - - Rede Elétrica

Figura IV.1- Unidade Experimental

A unidade experimental, mostrada na figura IV.1, foi dividida em dois sistemas. O primeiro formado pelo tanque 1 e demais equipamentos e o segundo formado pelos tanques 1 e 2 acoplados e demais equipamentos. A distinção entre os sistemas é feita abrindo-se ou fechando-se a válvula manual VM3. Mantendo-se esta válvula aberta, temos o sistema-*tanques acoplados*, caso contrário, temos o sistema-*tanque simples*.

IV.2 - Sistema de controle

A figura IV.1 mostra também a instrumentação para que a malha de controle seja viável: um sensor de pressão diferencial marca Contrisul, faixa de trabalho de 0 a 400 mbar. A comunicação entre os sinais analógicos da planta e o microcomputador, um equipamento digital, foi realizada por uma placa do tipo AD/DA (Analógico-Digital/Digital-Analógico) de 10 bits, marca Taurus com 8 canais de leitura na faixa de 1 a 5 volts, 2 canais de saída em corrente na faixa de 4 a 20 mA e 8 canais de saída na faixa de 1 a 5 volts, sendo programados por interrupções.

A rotina em tempo real que comanda o acionamento pelo microcomputador da placa AD/DA foi programada em Pascal para trabalhar por interrupções. A versão básica desta rotina, bem como sua interface gráfica, foram cedidas pelo laboratório de Controle e Microinformática do Departamento de Engenharia Elétrica da Universidade Federal de Santa Catarina.

O sistema utilizado chamado de Sistema de Desenvolvimento de Controladores Adaptativos, SDCA, é um ambiente de *software* em tempo real, que permite testar e comparar diferentes estratégias de controle adaptativo. Sua estrutura está baseada em técnicas de engenharia de *software* que facilitam o desenvolvimento integrado dos módulos e a sua manutenção. O programa possui uma estrutura modular, onde cada módulo foi definido levando-se em conta os critérios de coesão e acoplamento, reduzindo a sua complexidade e,

consequentemente, diminuindo a possibilidade de erros nas suas interfaces. A implementação modularizada permite, por exemplo, utilizar outras placas AD/DA, substituindo o módulo AD-DA, ou implementar diferentes controladores adaptativos ou não como módulos independentes, selecionáveis via teclado. Permite ainda, a visualização gráfica e o armazenamento das respostas do sistema em tempo real.

A ação de controle determinada pelo microcomputador é implementada através da manipulação da abertura da válvula de controle. Para ambos os sistemas, *tanque simples* e *tanques acoplados*, utilizou-se uma válvula pneumática de marca Hiter, modelo 201, com característica de igual percentagem e do tipo ar-abre. Para fazer a comunicação entre o microcomputador e a válvula utilizou-se um conversor eletro-pneumático, de marca Helix, com faixa de trabalho de 4 a 20 mA de entrada e saída de 3 a 15 psia.

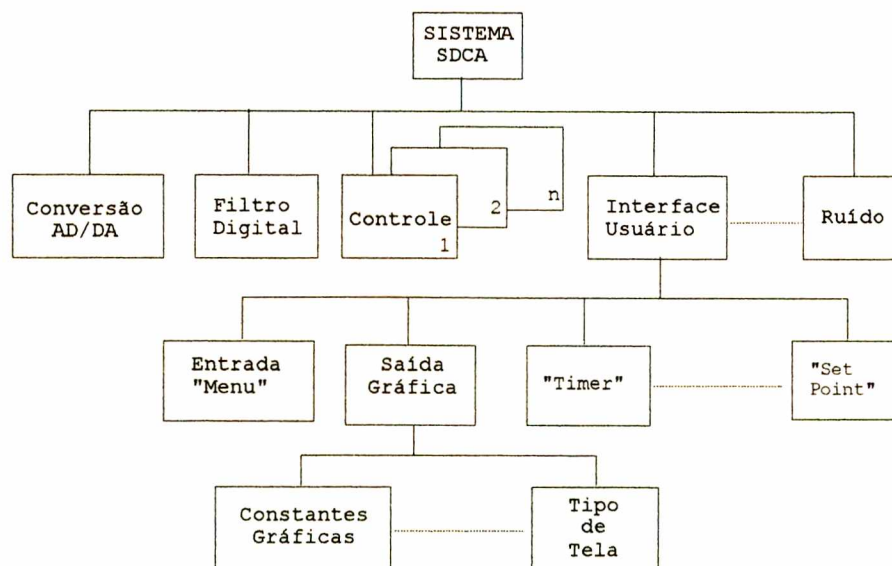


Figura IV 2 - Diagrama de Blocos do SDCA.

A água utilizada para a alimentação dos tanques é disponível através da rede hidráulica do laboratório, sendo fornecida por um tanque reservatório com capacidade para 157 litros e

bombeada por uma bomba centrífuga de marca Schneider, potência 1/4 de CV. O ar comprimido utilizado pelo conversor eletro-pneumático está disponível na rede pneumática do laboratório e é fornecido por um compressor de ar a uma pressão de 30 psia. Instalou-se um filtro regulador de pressão na entrada desse conversor para preservá-lo do óleo e da água que podem estar presentes no ar comprimido. As especificações de cada um dos equipamentos utilizados estão descritas no Apendice A.

A malha de controle implementada é mostrada abaixo:

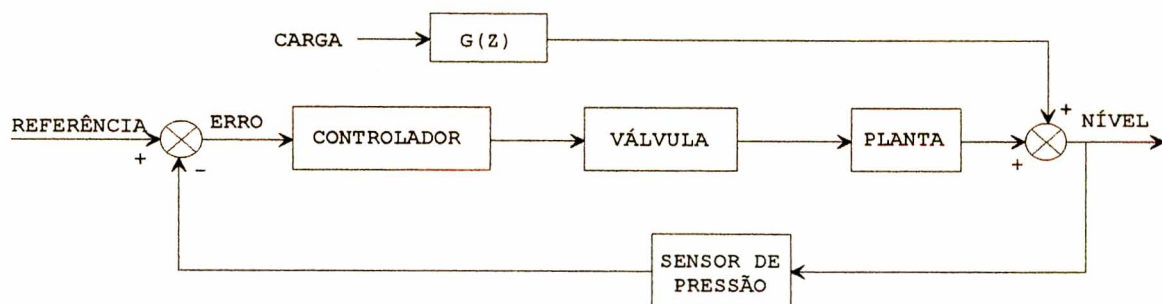


Figura IV 3 - Diagrama de Blocos em Malha Fechada

No diagrama de blocos mostrado na figura IV.3, o bloco "planta" representa tanto o sistema-tanque simples quanto o sistema-tanques acoplados.

IV.3 - Elaboração do *software* de Redes Neurais

O *software* de redes neurais desenvolvido no primeiro estágio deste trabalho foi escrito em linguagem de programação FORTRAN e segue as definições básicas descritas no item "Implementação de redes Neurais", do capítulo III.

O método de aprendizagem utilizado foi o *backpropagation*, juntamente com a estratégia de otimização *steepest descent*. As equações concernentes ao método são mostradas no capítulo III.

IV.4 - Definição do tamanho da rede

O número de neurônios na camada de entrada e de saída foi definido pela equação de diferenças, escolhida para representar os sistemas, equação III.18. Nesta equação as variáveis consideradas importantes são $y(k)$, $y(k-1)$, $u(k)$ e $y(k+1)$. Desta forma, como mostra a figura III.6 tem-se na primeira camada, 3 (três) neurônios, na última 1 (um) para a predição de apenas 1 (um) passo no futuro.

O número de neurônios escondidos foi definido como 6 (seis). Para chegar a este valor, utilizou-se o método de validação cruzada. Iniciou-se os treinos com apenas 1 (um) neurônio escondido e foi-se incrementando este número até que a convergência mostrou-se satisfatória tanto para o conjunto de treino quanto para o de teste.

IV.5 - Obtenção dos dados para treinamento e teste

Em malha aberta, os sistemas foram submetidos a variações na vazão de entrada, através de mudanças na abertura da válvula de controle. Os dados obtidos foram divididos em duas categorias: padrões para treinamento e padrões para teste.

Cada padrão é composto de dois conjuntos, um de entradas e um de saídas desejadas que serão alimentadas à rede. O primeiro conjunto é um vetor definido pelo número de neurônios na camada de entrada, o segundo, pelo número de neurônios na camada de saída.

O conjunto total de padrões de entrada, pode ser definido como uma matriz X_m com dimensão $(n_p \times n_{ne})$ e o de saída como uma matriz Y_m com dimensão $(n_p \times n_{ns})$. A matriz X_m representa as entradas da rede e a matriz Y_m as respectivas saídas reais ou alvos.

$$X_m = \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,n_{ne}} \\ x_{2,1} & x_{2,2} & x_{2,n_{ne}} \\ \vdots & \vdots & \vdots \\ x_{n_p,1} & x_{n_p,2} & x_{n_p,n_{ne}} \end{bmatrix} (n_p \times n_{ne}) \quad \text{IV.1}$$

$$Y_m = \begin{bmatrix} y_{1,1} & y_{1,2} & y_{1,n_{ns}} \\ y_{2,1} & y_{2,2} & y_{2,n_{ns}} \\ \vdots & \vdots & \vdots \\ y_{n_p,1} & y_{n_p,2} & y_{n_p,n_{ns}} \end{bmatrix} (n_p \times n_{ns}) \quad \text{IV.2}$$

IV.6 - Obtenção dos padrões de treinamento a partir dos dados brutos.

Os dados brutos são coletados aos pares, ou seja, a cada instante k de amostragem, são lidos e armazenados valores de $u(k)$ e $y(k)$, como mostrado na tabela IV.1:

Tabela IV.1 - Dados Genéricos Amostrados da Planta

k	$y(k)$	$u(k)$
1	$y(1)$	$u(1)$
2	$y(2)$	$u(2)$
3	$y(3)$	$u(3)$
4	$y(4)$	$u(4)$
\vdots	\vdots	\vdots

O primeiro padrão de treinamento é obtido a partir dos dados frizados na tabela IV.2:

Tabela IV.2 - Primeiro Padrão de Treinamento

k	y(k)	u(k)
1	y(1)	u(1)
2	y(2)	u(2)
3	y(3)	u(3)
4	y(4)	u(4)
:	:	:

Obtem-se, desse modo, o primeiro padrão de treinamento para a rede mostrada na figura III.6:

$$y(k) = y(2)$$

$$y(k-1) = y(1)$$

$$u(k) = u(2)$$

$$y(k+1) = y(3)$$

O padrão seguinte é obtido movendo-se a janela de dados mostrada na tabela IV.2, em um passo adiante, ou seja, um período de amostragem a frente, como pode ser observado na tabela IV.3

Tabela IV.3 - Segundo Padrão de Treinamento

k	y(k)	u(k)
1	y(1)	u(1)
2	y(2)	u(2)
3	y(3)	u(3)
4	y(4)	u(4)
:	:	:

Dessa forma, o próximo padrão será:

$$y(k) = y(3)$$

$$y(k-1) = y(2)$$

$$u(k) = u(3)$$

$$y(k+1) = y(4)$$

Esta janela é movida como descrito, até o final do conjunto de dados brutos, gerando assim, os padrões de entrada e saída a serem apresentados à rede. O tamanho dessa janela é função apenas do número de neurônios das camadas de entrada e de saída.

IV.7 - Adimensionalização dos padrões de entrada e saída

Após definidas as matrizes X_m e Y_m , seus componentes foram colocados na forma adimensional e em valores entre 0 e 1. Esta estratégia foi usada para que as entradas apresentadas à rede tivessem uma distribuição uniforme e que suas respectivas saídas ficassem dentro da faixa de saída da própria rede, lembrando que as saídas da rede são geradas pela função sigmoideal, cujo intervalo também é de 0 a 1.

Assim, a variável y , nível de água, foi adimensionalizada conforme a equação IV.3, onde os valores máximo e mínimo de y são respectivamente 2,00 e 0,00 metros e u , abertura da válvula, conforme a equação IV.4 onde os valores máximo e mínimo de u são 5,0 e 1,0 volts, respectivamente.

$$y = \frac{y - y_{\min}}{y_{\max} - y_{\min}} \quad \text{IV.3}$$

$$u = \left(\frac{u - u_{\min}}{u_{\max} - u_{\min}} \right) 0,6 + 0,2 \quad \text{IV.4}$$

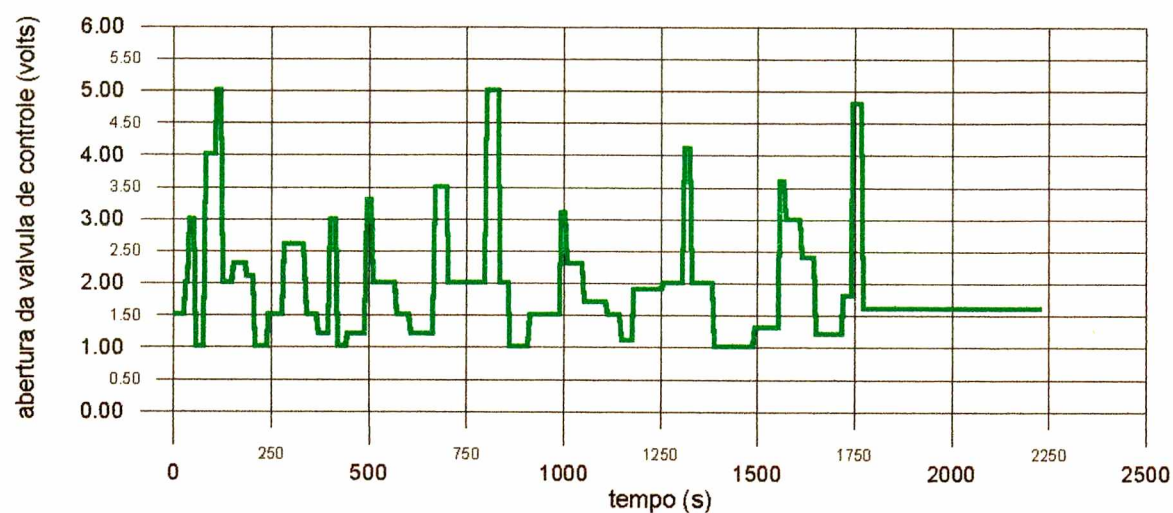
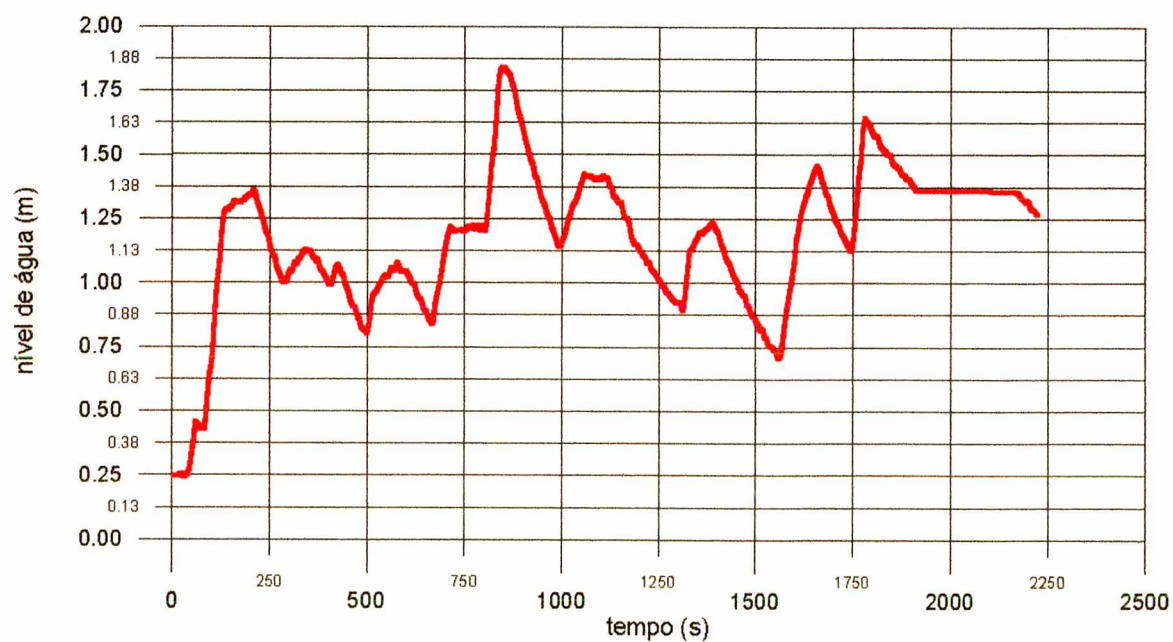
CAPÍTULO V

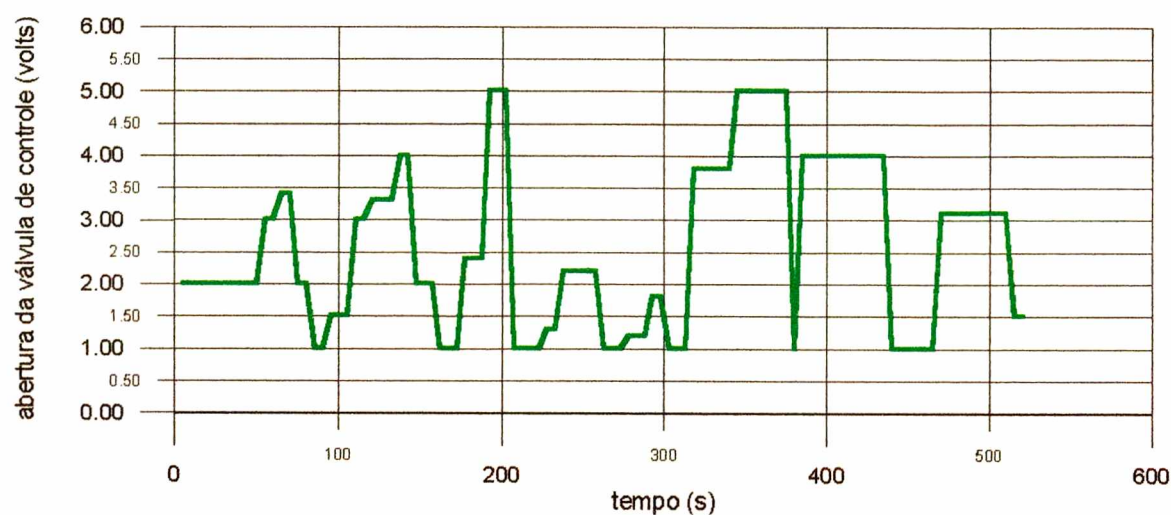
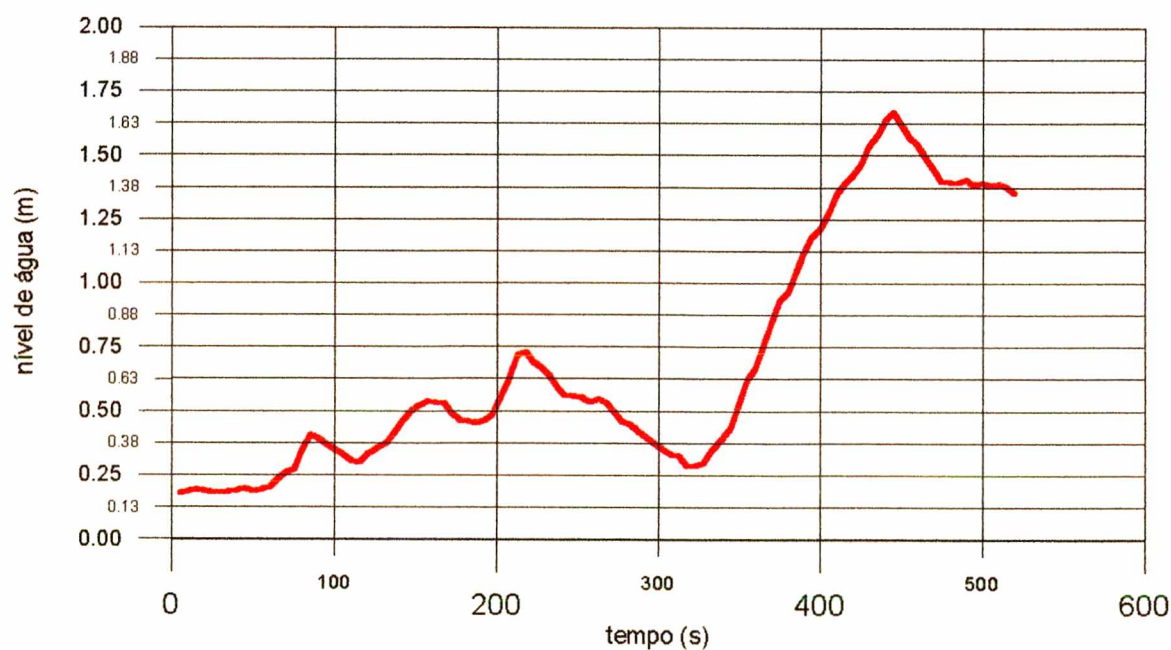
RESULTADOS E DISCUSSÕES

Neste capítulo apresenta-se os resultados de modelagem e controle utilizando redes neurais artificiais para um *tanque simples* e dois *tanques acoplados*.

V.1 - Modelagem dos sistemas usando Redes Neurais Artificiais -treinamento e testes

Os sistemas *tanque simples* e *tanques acoplados* foram modelados utilizando redes neurais artificiais. Dessa forma, realizou-se treinos e testes para ambos os sistemas. Para a realização de testes e treinos foram coletados dados de vazão de alimentação e de nível de água nos tanques. Para tanto, os sistemas foram excitados, variando-se a vazão de alimentação, frizando-se que os valores de vazão não são diretamente avaliáveis, dessa forma, utiliza-se a variável abertura da válvula, dada em volts. Assim, para válvula totalmente aberta, tem-se vazão máxima e sinal de 5,0 volts e para válvula totalmente fechada, vazão mínima e sinal de 1.0 volt. A altura máxima dos tanques é de 2,00 metros. Em ambos os sistemas usou-se um intervalo de amostragem de 5 segundos. O *software* usado para coletar os dados foi o SDCA, o qual foi descrito no capítulo IV. Os dados coletados são mostrados as figuras V.1 (a) a V.4 (b).

Figura V.1 (a) - variação na abertura da válvula - *tanque simples* - 390 pontosFigura V.1 (b) - variação no nível de água - *tanque simples* - 390 pontos

Figura V.2 (a) - variação na abertura da válvula - *tanque simples* - 100 pontosFigura V.2 (b) - variação no nível de água - *tanque simples* - 100 pontos

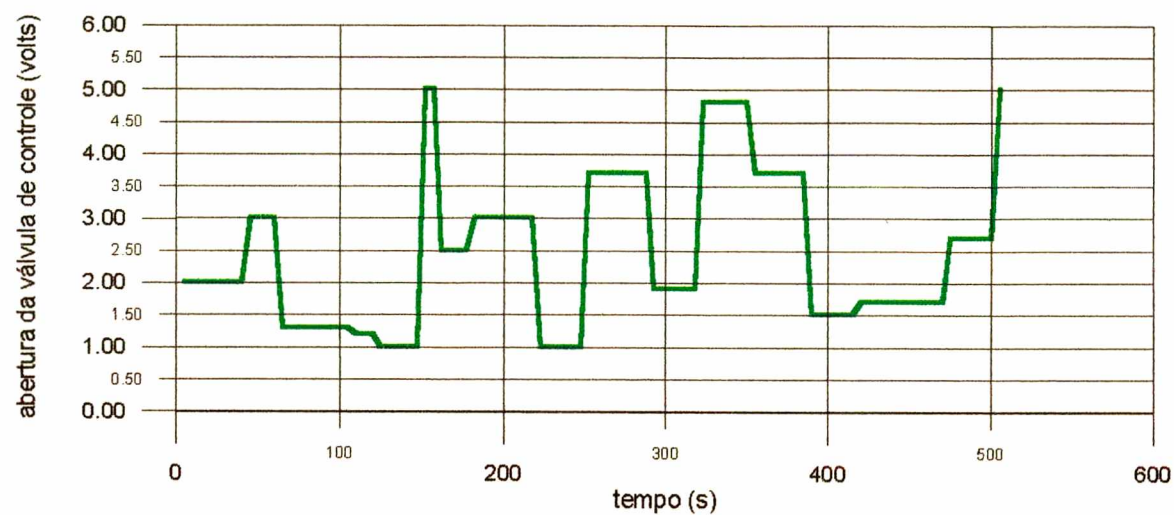


Figura V.3 (a) - variação na abertura da válvula - *tanques acoplados* - 100 pontos

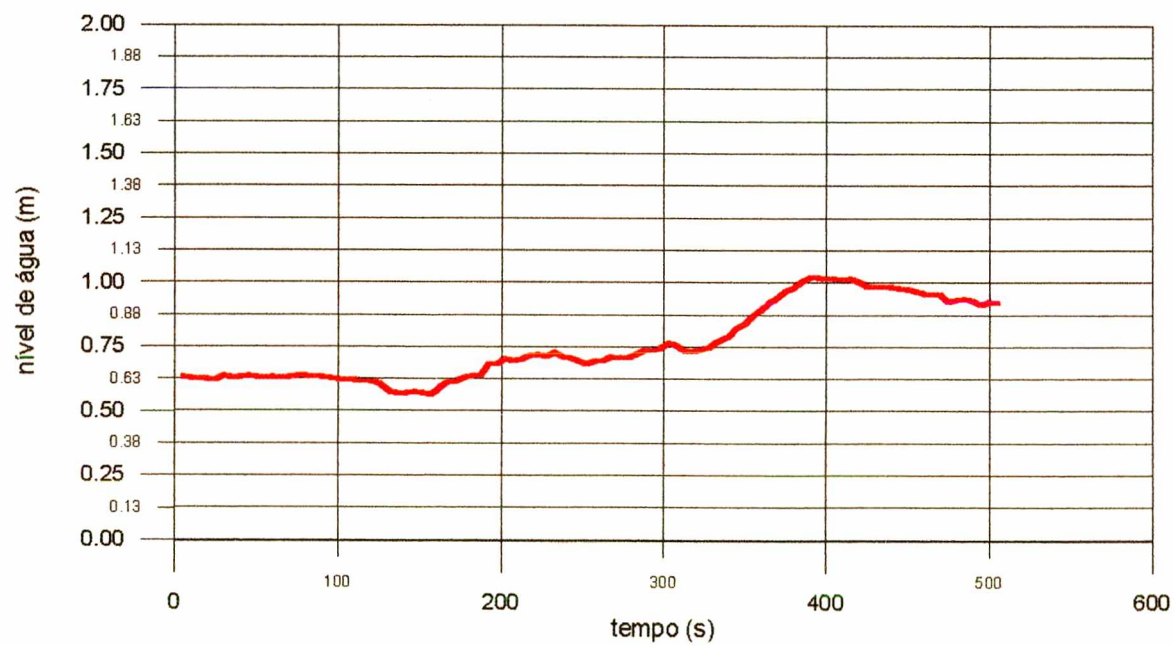
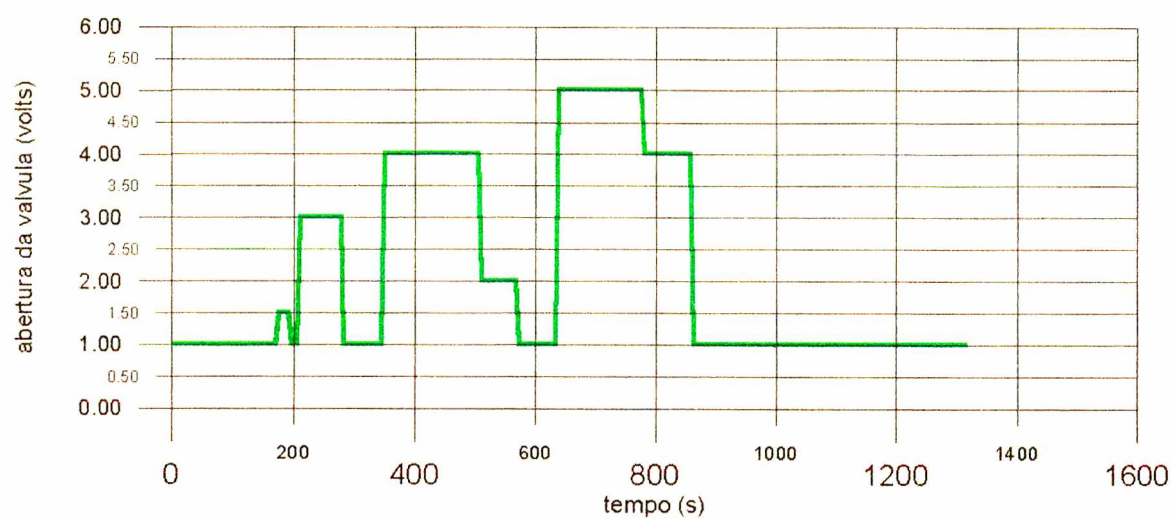
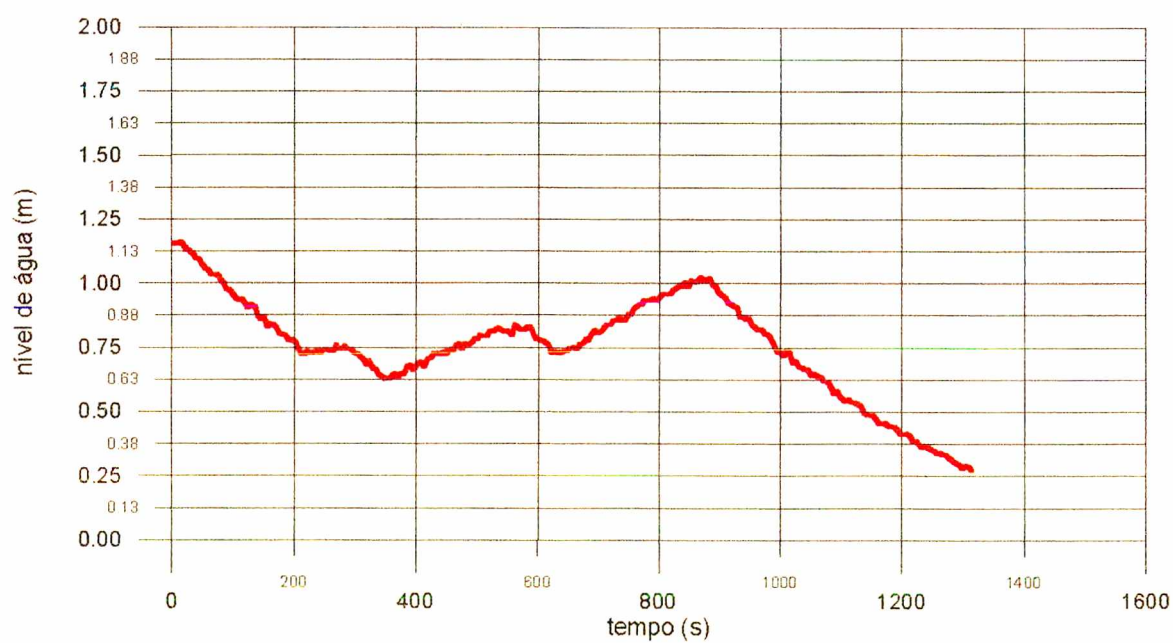


Figura V.3 (b) - variação no nível de água - *tanques acoplados* - 100 pontos

Figura V.4 (a) - variação na abertura da válvula - *tanques acoplados* - 100 pontosFigura V.4 (b) - variação no nível de água - *tanques acoplados* - 100 pontos

O sistema *tanque simples* sofreu mudanças na sua dinâmica pois ocorreram modificações na posição da haste da válvula, devidas à manutenção e reparos da mesma. Os dados mostrados nas figuras V.1 (a) e V.1 (b) são referentes ao sistema antes das modificações e foram utilizados apenas para modelagem e não para o controle do processo. São mostrados para evidenciar a flexibilidade da rede usada e a facilidade de se refazer um treino e de obter-se novos parâmetros para a mesma.

V.1.1 - Sistema *Tanque Simples*

Este tanque, com altura de 2,00 metros possui parâmetros variantes com o ponto de operação. Este sistema de primeira ordem apresenta características suficientemente não lineares para que seu estudo se torne atraente no âmbito de redes neurais artificiais.

V.1.1.1 - Treinamento e testes - sistema *Tanque Simples*

Realizaram-se 2 treinos para este sistema. Para compor o primeiro conjunto de treinamento foram usados os 100 primeiros pontos de $y(k)$ e $u(k)$ mostrados nas figuras V.1 (a) e V.1 (b). Os demais pontos destas figuras foram utilizados para testar os parâmetros (pesos), obtidos no treino da rede. Para compor o segundo conjunto de treinamento, usou-se os pontos das figuras V.2 (a) e V.2 (b). Não foram realizados testes *off line* para este caso. Dos dados obtidos no 'treino 2' partiu-se diretamente para o controle. Dessa forma, os testes foram realizados *on line* e são mostrados posteriormente neste capítulo.

Em cada treino e teste, os valores de $y(k)$ e $u(k)$ foram adimensionalizados e rearranjados para formar as matrizes de padrões de entrada e de saídas alvo, como descrito no capítulo IV. Para cada treino, a matriz de padrões de entrada, X_m , possui a dimensão (98 x 3) e a de padrões

de saída, Y_m , (98×1) . Para o teste, X_m tinha dimensão de (288×3) e Y_m , (288×1) . Estas dimensões foram definidas pelo número de padrões usados em cada treino, $n_p=98$, número de neurônios de entrada, $n_{ne}=3$, neurônios na camada de saída, $n_{ns}=1$, e número de padrões usados no teste, $n_p = 288$.

V.1.1.2 - 'Treino 1' - *Tanque Simples*

Foram necessárias 18.000 apresentações do conjunto de treinamento para que a rede neural fosse capaz de aprender o conjunto de padrões apresentados e para que se atingisse um valor satisfatório para o erro absoluto entre a sua saída e a saída real. Este erro era considerado satisfatório se sua média fosse menor que 5% . Foram estimados um total de 18 parâmetros. Partiu-se de um conjunto de pesos aleatórios, distribuídos entre -1 e 1 e chegou-se aos valores mostrados nas tabelas abaixo:

Tabela V.1 (a) - Valores finais obtidos para os pesos entre as camadas intermediária e de saída - 'treino 1' - *tanque simples*

neurônio da camada 2	neurônio da camada 3	peso $w_{2i,k}$
1	1	-0,4461
2	1	0,9338
3	1	0,3436
4	1	1,2110
5	1	1,3886
6	1	-8,5668

Tabela V.1 (b) - Valores finais obtidos para os pesos entre as camadas de entrada e intermediária - 'treino 1' - *tanque simples*

neurônio da camada 1	neurônio da camada 2	peso $w_{1,i}$
1	1	0,1423
1	2	0,5452
1	3	0,6399
1	4	0,5478
1	5	0,5293
1	6	-2,2271
2	1	0,1449
2	2	0,5765
2	3	0,6314
2	4	0,5600
2	5	0,5233
2	6	0,2967
3	1	0,1448
3	2	0,5901
3	3	0,6902
3	4	0,5578
3	5	0,5171
3	6	0,0233

Para a obtenção destes pesos a rede usada possuía a seguinte configuração inicial:

$n_{ne} = 3$, $n_{ni} = 6$, $n_{ns} = 1$ e pesos iniciais aleatórios distribuídos entre -1 e 1. O valor escolhido para a taxa de aprendizagem, η , usada no algoritmo *backpropagation* foi 0.06.

A figura V.5 mostra o resultado para este primeiro treino:

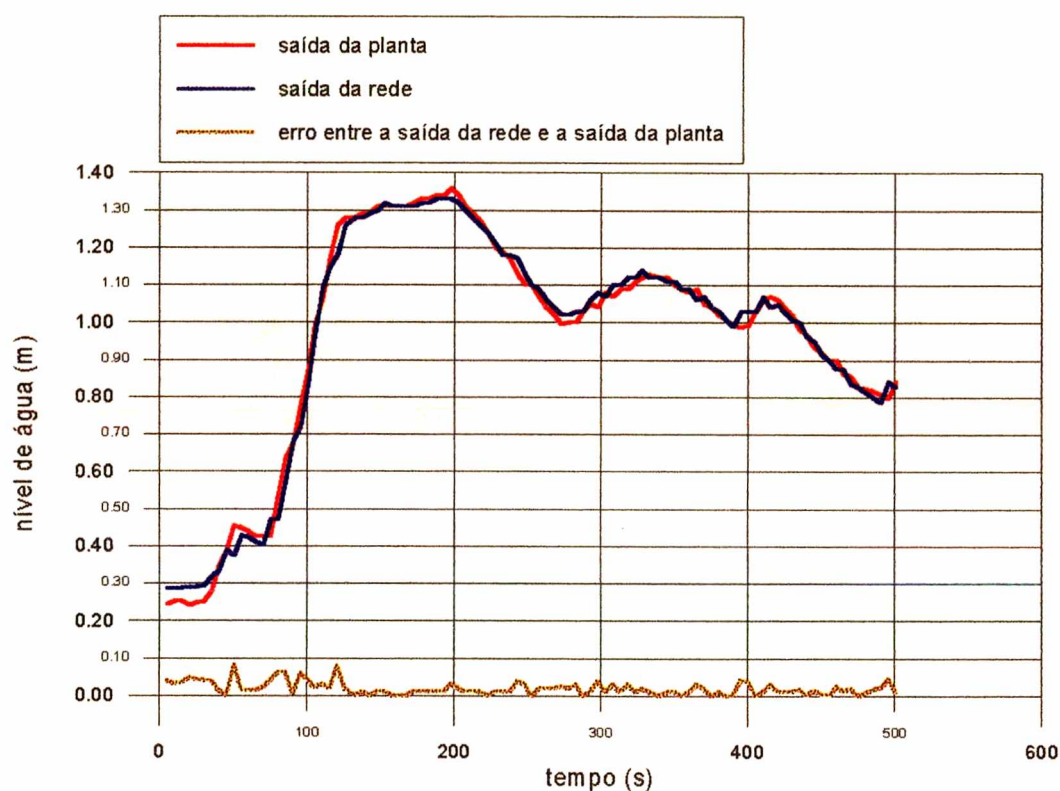


Figura V.5 - Rede Neuronal Artificial - 'treino 1' - *tanque simples*

V.1.1.3 - Teste referente ao 'Treino 1' - *Tanque Simples*

Realizou-se apenas 1 teste com um conjunto de 290 pontos, retirados das figuras V.1 (a) e V.1 (b). Utilizou-se para tanto os pesos mostrados nas tabelas V.1 (a) e V.1 (b). Nesta fase a rede é usada de maneira *feedforward*, ou seja, apresentou-se os padrões de entrada e a rede forneceu a saída com base nos parâmetros obtidos no 'treino 1'. O resultado é mostrado a seguir:

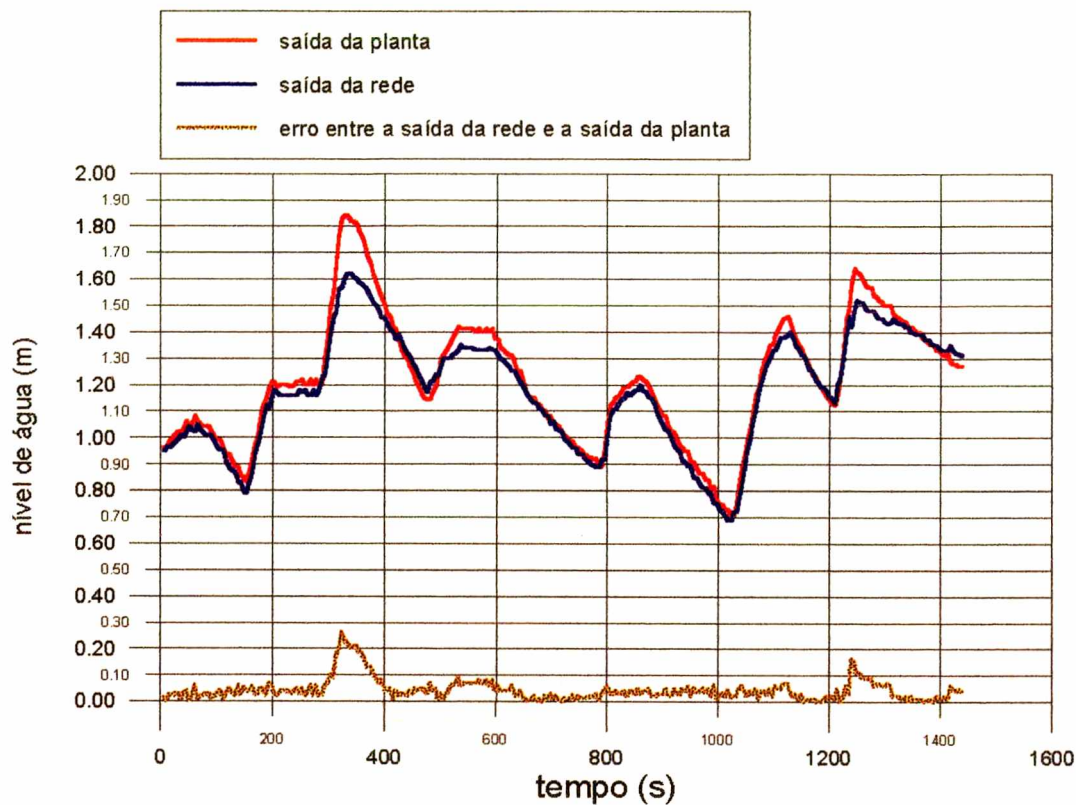


Figura V.6 - Rede Neural Artificial - 'teste 1' referente ao 'treino 1' - *tanque simples*

V.1.1.4 - Comentários

Observa-se, no 'treino 1', figura V.5, desvios pequenos entre a resposta da rede e a saída da planta, evidenciando que a rede 'aprendeu' o conjunto de treinamento que lhe foi apresentado como alvo.

A figura V.6, mostra o teste realizado para verificar a capacidade de generalização da rede configurada no 'treino 1'. Esta rede, já completamente definida, apresenta como parâmetros internos os pesos mostrados nas tabelas V.1 (a) e V.1 (b). Nos testes, como apresentado no capítulo III, os pesos são mantidos fixos. O conjunto de padrões de entrada é apresentado e a rede fornece a saída. Na figura V.6 são mostradas as curvas correspondentes a saída da planta e a saída da rede. Observa-se desvios maiores do que aqueles do 'treino 1'. Estes desvios são maiores

para valores de $y(k)$ superiores a 1,40 metros. Isto pode ser explicado, primeiramente porque a rede adaptou-se de forma a simular o conjunto específico de padrões de treinamento e segundo, pelo fato de que a rede foi treinada para valores de $y(k)$ variando entre 0,20 e 1,40 metros. Porém, ainda assim, a rede foi capaz de acompanhar o comportamento do sistema para todo o conjunto de teste apresentado.

V.1.1.5 - 'Treino 2' - *Tanque Simples*

Para este treino, ainda referente ao *tanque simples*, utilizou-se os mesmos critérios e metodologia usados no 'treino 1'. Neste caso foram necessárias 12.000 apresentações do conjunto de treinamento retirado das figuras V.2 (a) e V.2 (b). A configuração da rede é a mesma usada no 'treino 1'. Foram estimados um total de 18 parâmetros para a rede. Seguem-se os valores obtidos para as conexões

Tabela V.2 (a) - Valores finais obtidos para os pesos entre as camadas intermediária e de saída - 'treino 2' - *tanque simples*

neurônio da camada 2	neurônio da camada 3	peso $w_{2i,k}$
1	1	0,1945
2	1	0,3462
3	1	0,1025
4	1	0,3711
5	1	2,8784
6	1	-8,5783

Tabela V.2 (b) - Valores finais obtidos para os pesos entre as camadas de entrada e intermediária - 'treino 2' - *tanque simples*

neurônio da camada 1	neurônio da camada 2	peso $w_{1i,i}$
1	1	0,4970
1	2	0,2838
1	3	0,3617
1	4	0,3598
1	5	0,4394
1	6	-2,8629
2	1	0,4774
2	2	0,2756
2	3	0,3788
2	4	0,3585
2	5	0,4350
2	6	0,9410
3	1	0,4601
3	2	0,2711
3	3	0,3747
3	4	0,3783
3	5	0,4263
3	6	0,1229

Optou-se por testar a rede configurada no 'treino 2', já atuando *on line* em situações de controle . Estes testes são apresentados em itens posteriores, neste capítulo.

O grafico V.7 mostra o resultado deste treino:

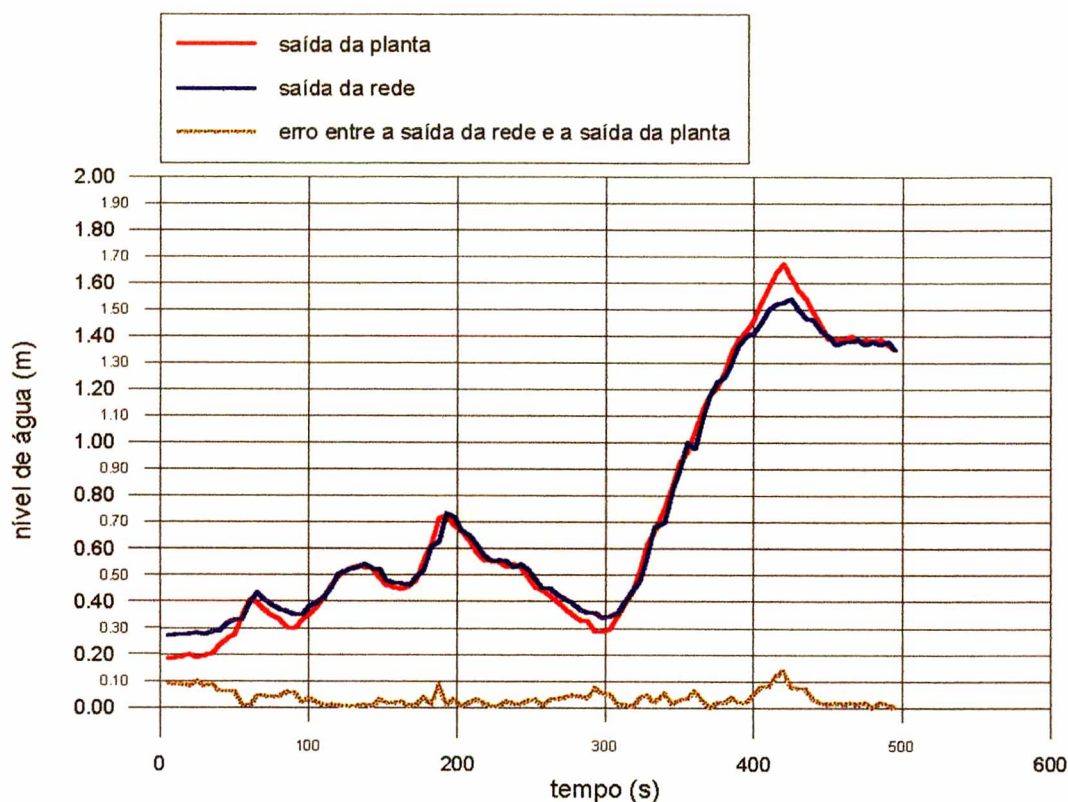


Figura V.7 - 'Treino 2' - *tanque simples*

V.1.1.6 - Comentários

Observa-se que há uma variação entre os valores dos pesos obtidos neste treino, (tabelas V.2 (a) e (b)), e dos obtidos no 'treino 1', (tabelas V.1 (a) e (b)). Isto pode ser explicado pela mudança ocorrida na dinâmica do sistema, devido à manutenção da válvula de controle, como já comentado neste capítulo. Nota-se, porém, uma tendência para os valores de alguns pesos. Isto indica que em ambos os treinos as principais características do sistema foram 'captadas'. Na figura V.6 pode-se observar que a saída da rede acompanha a saída do sistema com precisão para a maioria dos pontos. Os maiores desvios são encontrados no início do conjunto de dados e em torno do tempo de 400 segundos, onde as saídas da rede ficam abaixo da saída do sistema.

V.1.2 - Sistema *Tanques Acoplados*

Este sistema, de segunda ordem, apresenta uma dinâmica mais lenta que a do sistema *tanque simples*, porém suas características de comportamento e suas não linearidades são semelhantes.

V.1.2.1 - Treinamento e testes - sistema *Tanques Acoplados*

Para treinar a rede neuronal que simulasse o comportamento deste sistema foram utilizados os dados de $y(k)$ e $u(k)$ apresentados nas figuras V.3 (a) e V.3 (b). Já os dados das figuras V.4 (a) e V.4 (b) foram usados para testar a confiabilidade dos pesos obtidos neste treino.

Para ambos, treino e teste referentes a este sistema usou-se a metodologia descrita no capítulo IV e já utilizada para o sistema tanque simples. Da mesma forma, as matrizes X_m e Y_m possuem, respectivamente, as dimensões (98×3) e (98×1) para o treino e (236×3) e (236×1) para o teste. Para tanto foram usados 98 padrões de treinamento e 236 padrões de teste.

V.1.2.2 - Treino - *Tanques Acoplados*

Após 4000 apresentações do conjunto de treinamento, obteve-se os seguintes pesos apresentados nas tabelas V.3 (a) e (b). A figura V.8 mostra o resultado deste treino.

A configuração inicial da rede usada é a seguinte:

$n_{ni} = 3$, $n_{ne} = 6$, $n_{ns} = 1$ e pesos iniciais usados foram os obtidos no 'treino 2' referente ao sistema *tanque simples*. A taxa de aprendizagem η foi mantida em 0.06

Tabela V.3 (a) - Valores finais obtidos para os pesos entre as camadas intermediária e de saída - treino - *tanques acoplados*

neurônio da camada 2	neurônio da camada 3	peso $w_{2i,k}$
1	1	0,2157
2	1	0,3748
3	1	0,1277
4	1	0,3965
5	1	2,9019
6	1	-8,4951

V.1.2.3 - Teste - *Tanques Acoplados*

A figura V.9 mostra os resultados do teste realizado usando a rede configurada anteriormente e com pesos obtidos no treino para o sistema *tanques acoplados*.

Tabela V.3 (b) - Valores finais obtidos para os pesos entre as camadas de entrada e intermediária - treino - *tanques acoplados*

neurônio da camada 1	neurônio da camada 2	peso $w_{li,i}$
1	1	0.4933
1	2	0,2763
1	3	0,3593
1	4	0,3522
1	5	0,3853
1	6	-2,7704
2	1	0,4760
2	2	0,2724
2	3	0,3779
2	4	0,3553
2	5	0,4131
2	6	0,9427
3	1	0,4585
3	2	0,2779
3	3	0,3737
3	4	0,3750
3	5	0,4025
3	6	0,1491

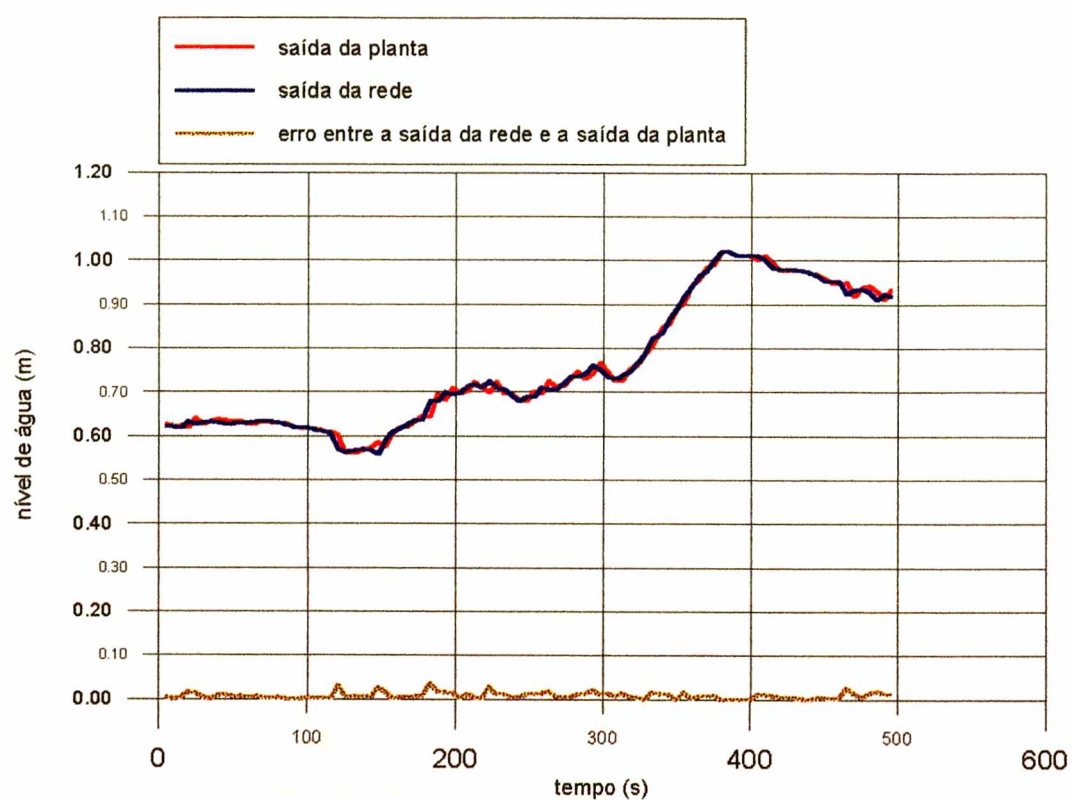


Figura V.8 - 'Treino' - tanques acoplados

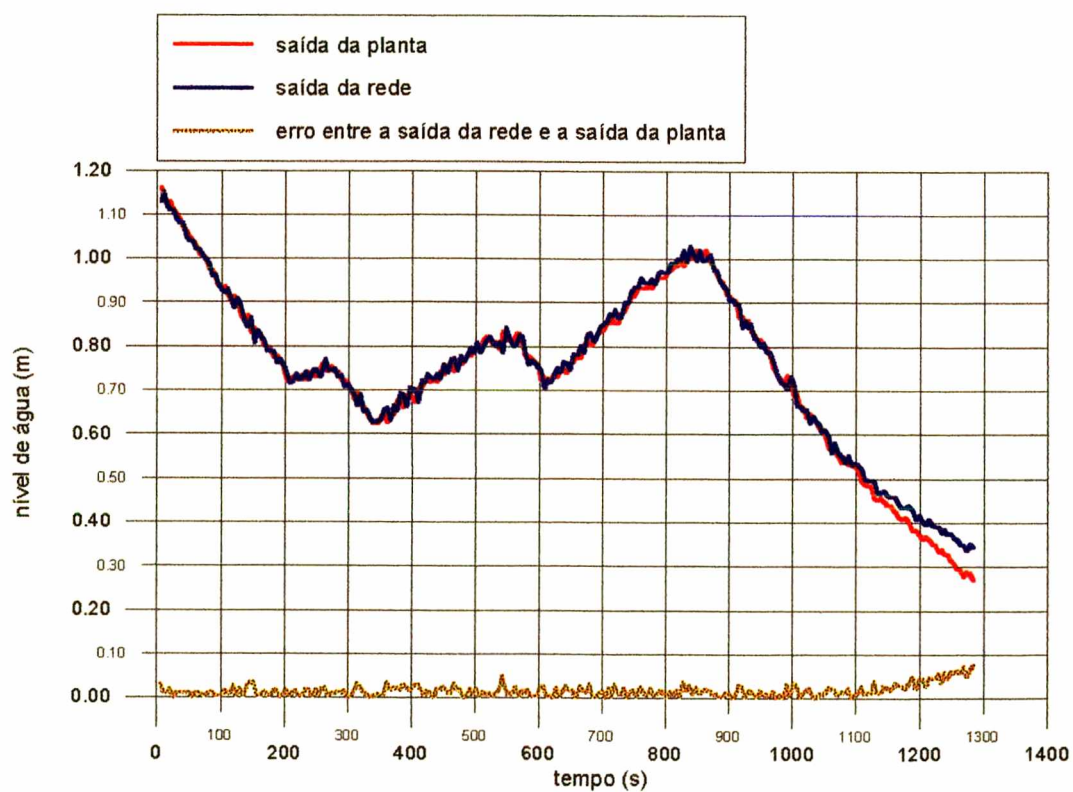


Figura V.9 - 'Teste' - tanques acoplados

V.1.2.4 - Comentários

As tabelas V.3 (a) e (b) mostram os pesos obtidos no treino da rede para o sistema *tanques acoplados*. Comparando-se estes valores aos das tabelas V.2 (a) e (b), observa-se que eles são quase idênticos. Isto pode ser explicado pelo fato de se ter usado os valores destas tabelas como pesos iniciais para o treino referente ao sistema *tanques acoplados*. Dessa forma o treino pode ser visto como tendencioso, porém optou-se pelo uso destes valores devido a semelhança entre os sistemas. A opção feita mostrou-se satisfatória, uma vez que o conjunto de treinamento, foi 'aprendido' com rapidez e erros pequenos foram observados, como mostra a figura V.8. O mesmo ocorrendo com o conjunto de testes, figura V.9.

V.2 - Controle

Terminados os treinos e testes, as redes neurais artificiais já completamente definidas em termos de número de neurônios e de parâmetros internos (pesos), foram utilizadas como modelos dos processos no esquema de controle preditivo descrito no capítulo III. A estratégia foi implementada como parte do *software* SDCA, ilustrado em diagrama de blocos no capítulo IV. Os resultados do controle do nível dos sistema *tanque simples* e *tanques acoplados* são mostrados a seguir.

V.2.1 - Sistema *Tanque Simples*

Este sistema, mostrado na figura IV.1, composto por uma tanque com altura de 2,00 metros, possui duas saídas. Na saída principal, a válvula manual-VM1 foi mantida em uma posição fixa durante todos os experimentos, o mesmo ocorrendo com a válvula manual-VM2, localizada na saída secundária. Esta saída possui também uma válvula solenóide-VS, de

acionamento manual. Nos testes de controle as perturbações foram geradas pela abertura da válvula VS. O aumento na vazão de saída com esta carga ficou em torno 10 %. A ação de controle foi implementada através da válvula pneumática-VP, acionada pelo microcomputador/controlador-MCC. As figuras seguintes mostram a resposta do sistema para mudanças no *set point* e para aplicação de carga.

A ação de controle é calculada pela equação III.42, deduzida a partir da função custo, equação III.20, onde $H=1$ e $NU=1$. O controlador obtido, utilizando estes valores, é dito **imediat**, pois tem o objetivo de levar a planta ao estado desejado já no próximo intervalo de amostragem, uma vez que nesta abordagem o modelo calcula a saída do processo um passo no futuro. O modelo é dado pela rede neural escolhida para representar o sistema, definida no 'treino 1' e cujos parâmetros são mostrados nas tabelas V.2 (a) e (b). A trajetória de referência é dada pelas equações III.21 e III.22. Os parâmetros do controlador foram ajustados de forma que sua performance global fosse satisfatória. O valor de λ foi mantido fixo em 0,0022 e variou-se α em uma faixa entre 0,60 e 0,90. Performance semelhante poderia ser obtida fixando-se α e variando-se λ . Optou-se pela primeira estratégia pois a faixa abrangida por α é menor que aquela abrangida por λ . O intervalo de amostragem utilizado foi de 5 segundos, que é o mesmo intervalo utilizado na coleta de dados para o treinamento da rede.

V.2.1.1 - Comentários

As figuras V.10 (a) até V.12 (b) mostram as variações no *set point* da variável nível de água, bem como o comportamento do sistema frente a perturbações do tipo carga. Mostram também a ação de controle implementada pela válvula pneumática. Na figura V.10 (a) observa-se que o controlador foi capaz de elevar o nível do tanque de 0,75 para 1,00 metro e de rejeitar a carga imposta. Na figura V.10 (b), mostra que para este caso a válvula foi mais solicitada para manter o nível a 1,00 metro. Isto pode ser explicado lembrando que os parâmetros do

sistema variam com o ponto de operação, e os parâmetros do controlador podem não estar adequados a esta variação.

As figuras V.11 (a) e (b) mostram que o controlador é capaz de elevar o nível do tanque e retornar ao ponto de operação inicial, mantendo um comportamento coerente.

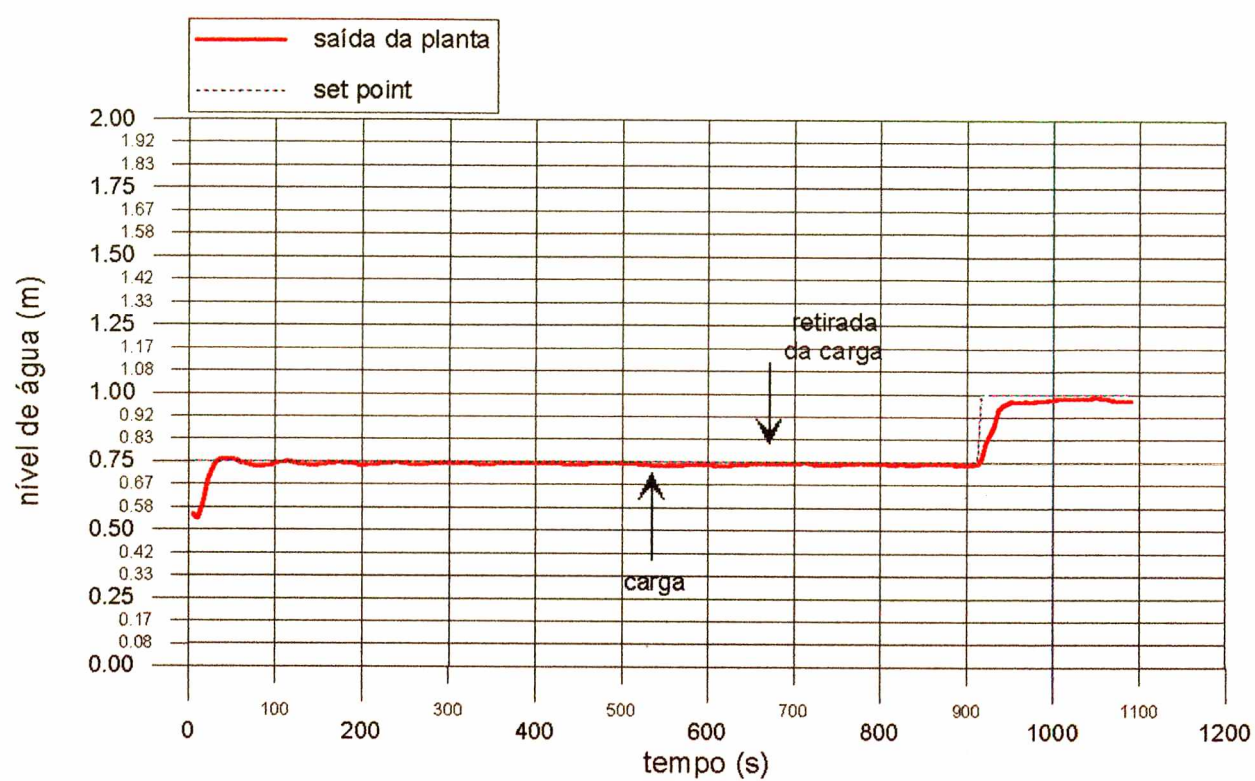
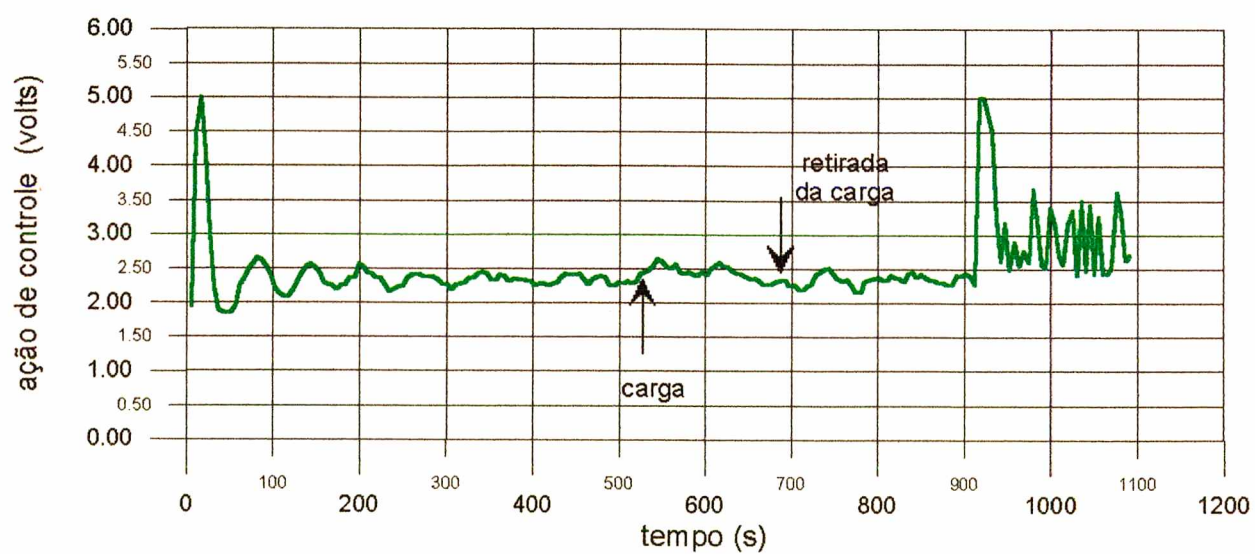
As figuras V.12 (a) e (b) evidenciam o comportamento diferenciado do controlador para diferentes pontos de operação. Para valores de *set point* entre 1,00 e 1,20 metros, a válvula de controle é mais exigida do que para pontos em torno de 0,60 metros, mas de forma geral, o controlador é capaz de levar o sistema ao estado desejado e mantê-lo sem grandes problemas.

V.2.2 - Comportamento da Rede Neural durante o controle do processo

As figuras V.13 até V.15 mostram a saída da rede e a saída do sistema, bem como a diferença entre ambas, ou seja, o erro de modelagem que ocorreu durante a implementação do sistema de controle baseado em uma rede neural artificial. Os dados apresentados nestas figuras são testes *on line* realizados com a rede configurada e apresentada em V.1.1.5. Deve-se lembrar que optou-se pela não realização de testes *off line* para aquela rede. O erro mostrado nas figuras é o erro absoluto entre a saída da planta e a saída da rede.

V.2.2.1 - Comentários

Como é possível observar nas figuras V.13 até V.15, o modelo gerado pela rede neural artificial acompanha o sistema de forma bastante precisa. O erro torna-se irrelevante comparado com os erros ocasionados por ruídos nas medidas.

Figura V.10 (a) - Mudança no *set point* e aplicação de carga - sistema *tanque simples*Figura 10 (b) - Ação de controle - sistema *tanque simples*

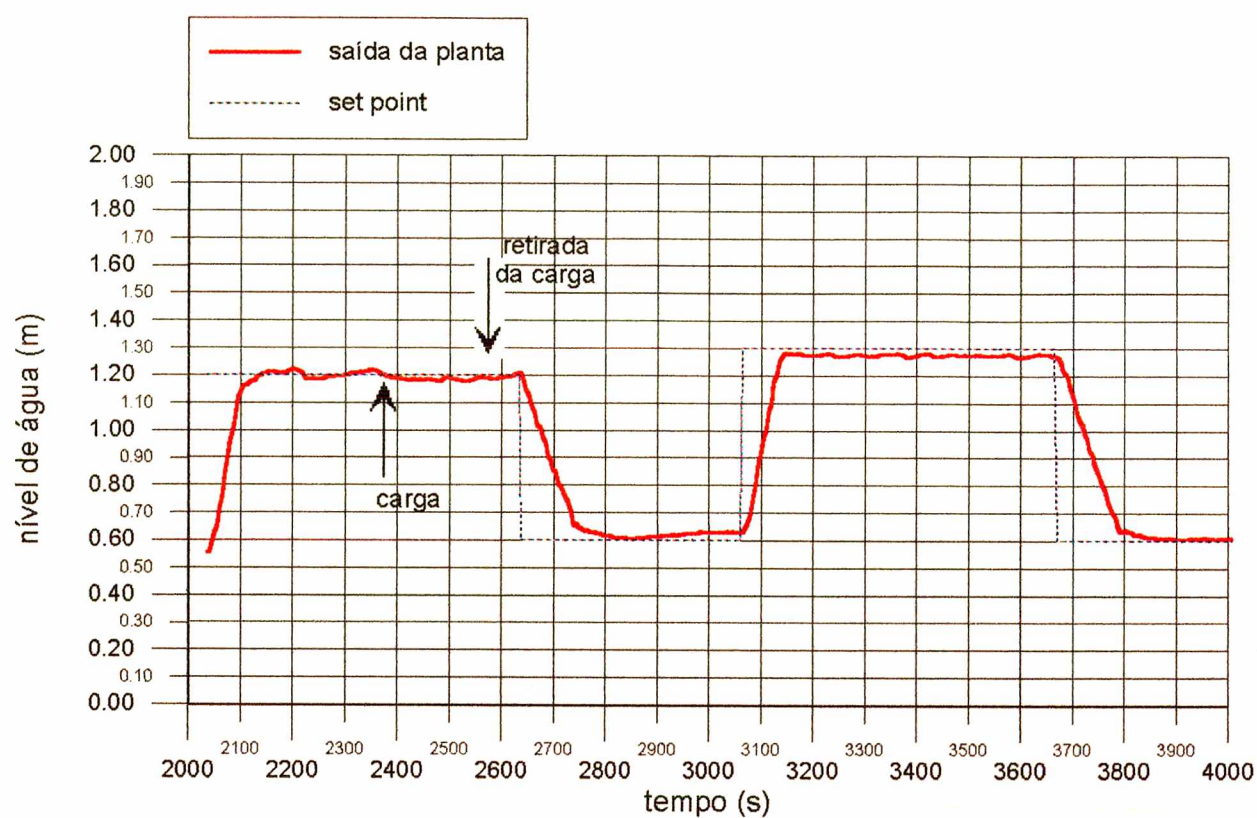


Figura V.11 (a) - Mudança no *set point* e aplicação de carga - sistema *tanque simples*

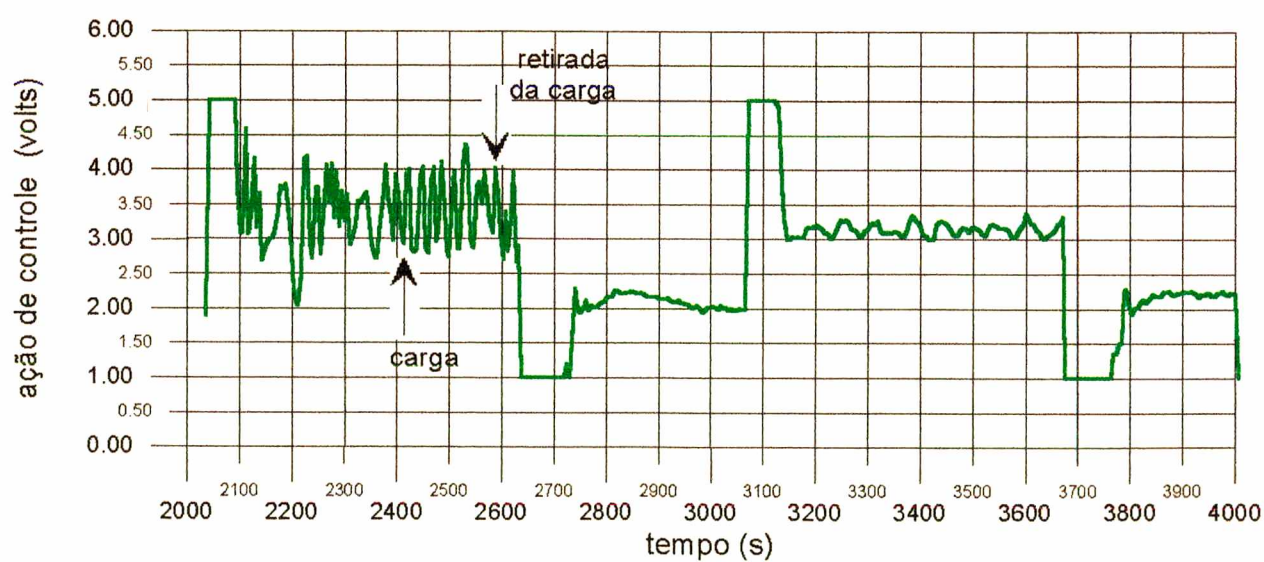


Figura 11 (b) - Ação de controle - sistema *tanque simples*



Figura V.12 (a) - Mudança no *set point* e aplicação de carga - sistema *tanque simples*

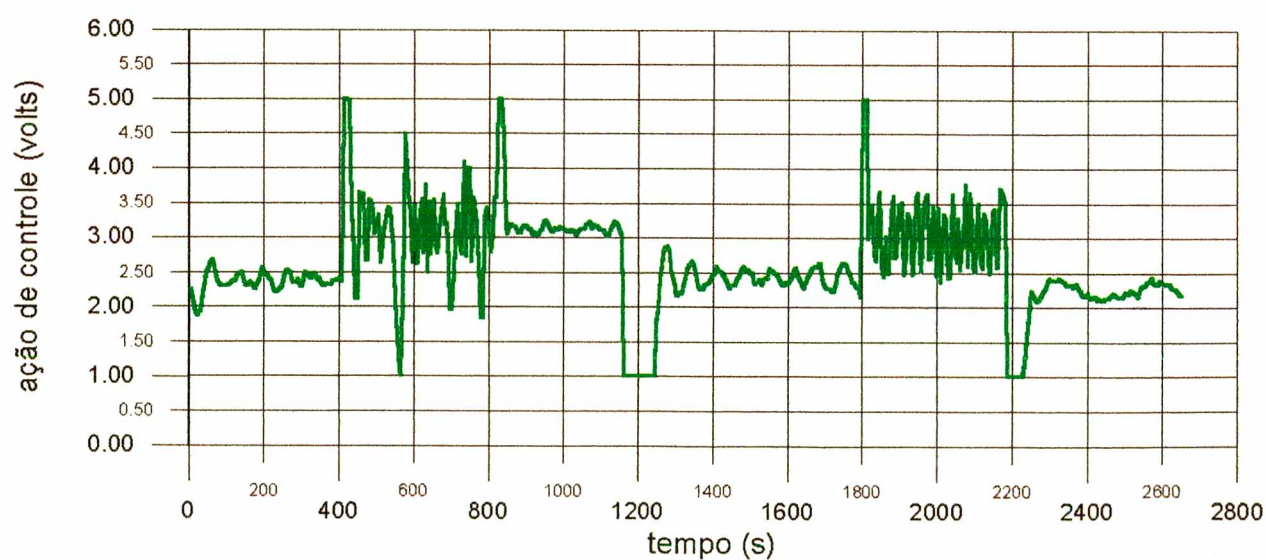
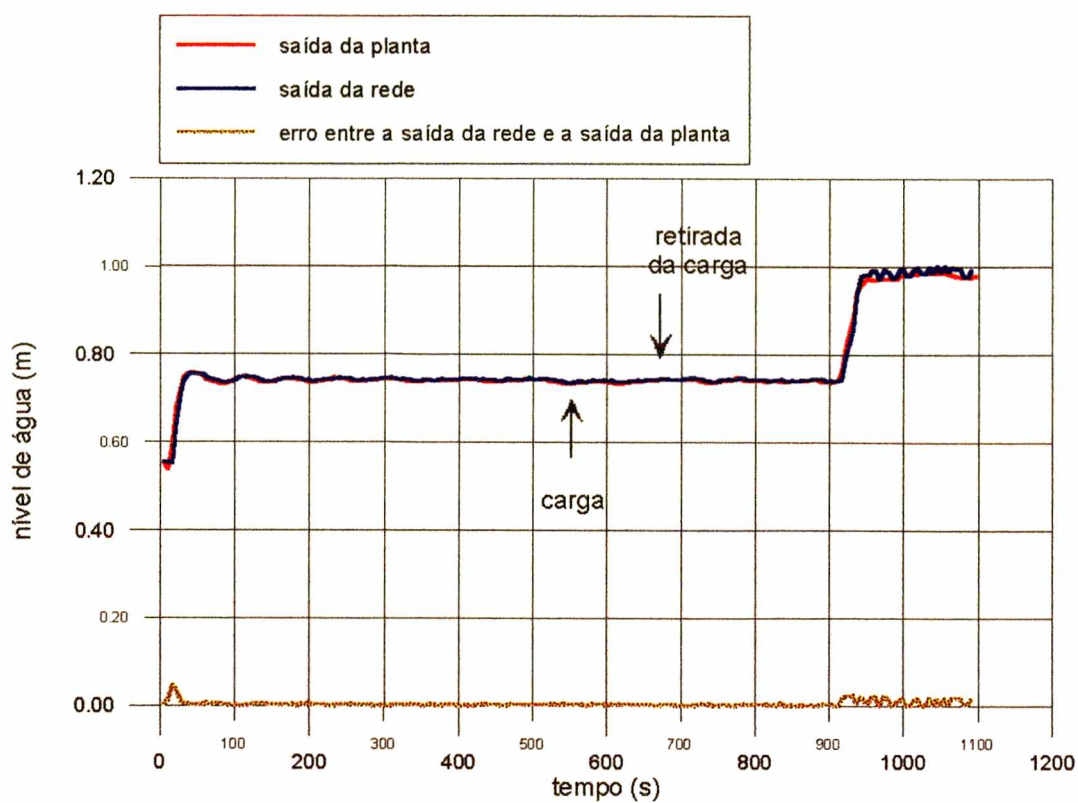
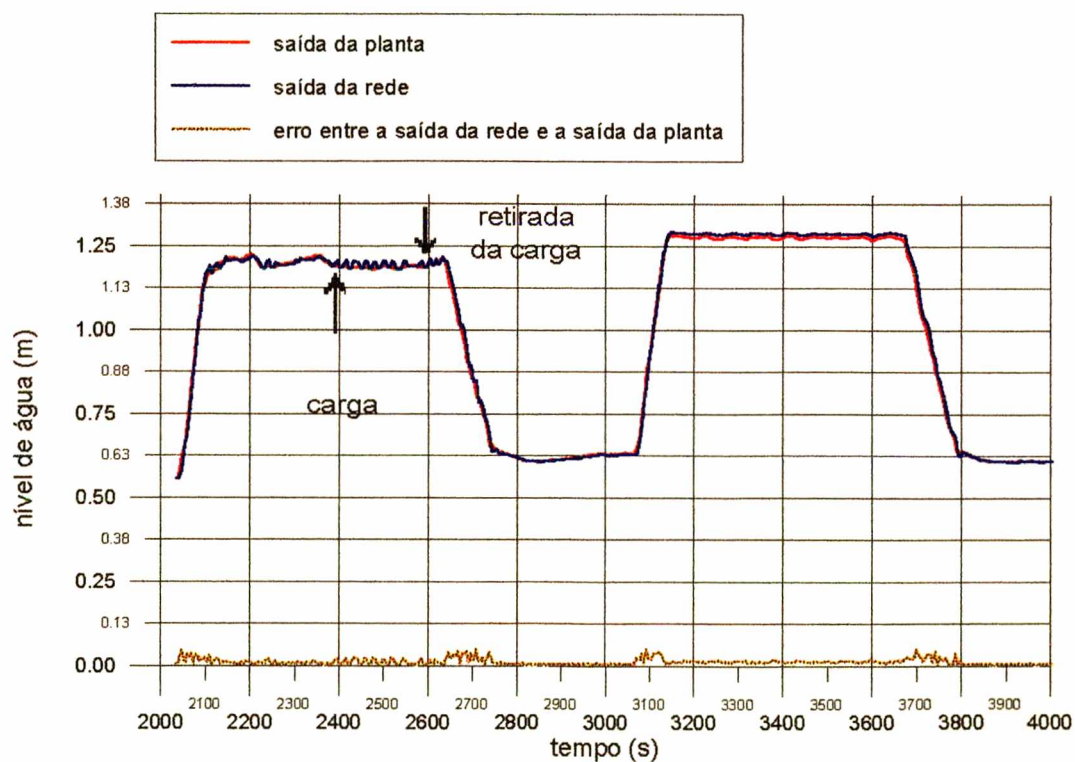


Figura 12 (b) - Ação de controle - sistema *tanque simples*

Figura V.13 - Comparação entre a saída da rede e a da planta - sistema *tanque simples*Figura V.14 - Comparação entre a saída da rede e a da planta - sistema *tanque simples*

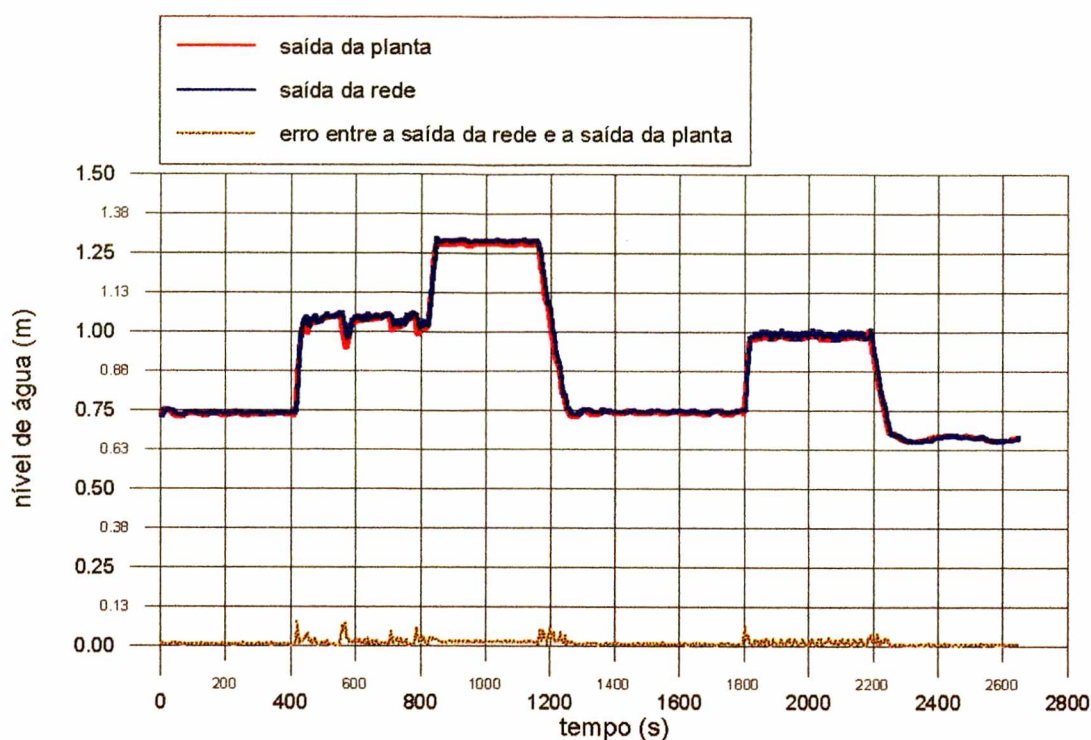


Figura V.15- Comparação entre a saída da rede e a da planta - sistema *tanque simples*

V.2.3 - Sistema *Tanques Acoplados*

Este sistema é mostrado na figura IV.1. Toda a instrumentação e as redes pneumática, elétrica e hidráulica são as mesmas do sistema *tanque simples* porém, agora, a válvula VM3 é mantida totalmente aberta para que se estabeleça a comunicação entre os tanques 1 e 2. A malha de controle também é a mesma. Para a implementação desta malha utilizou-se como variável medida o nível do tanque 1.

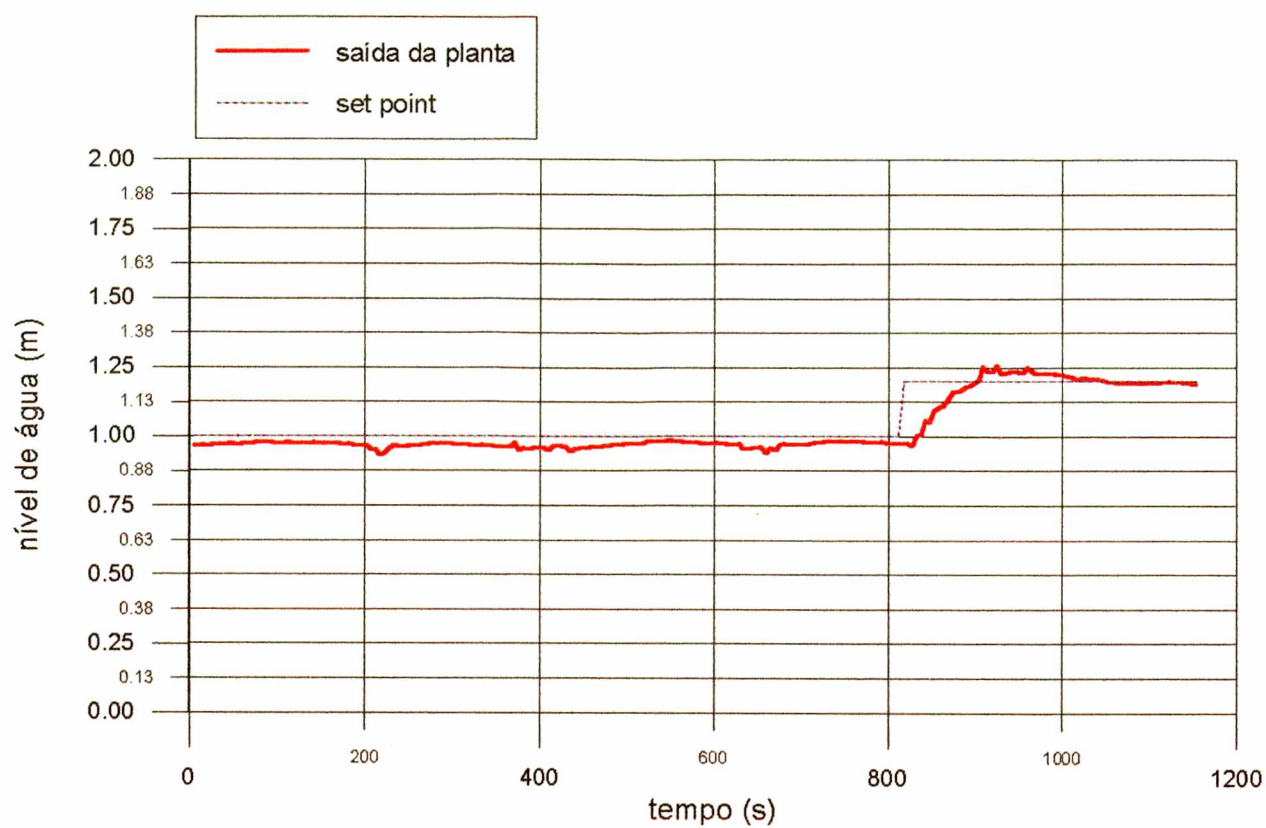
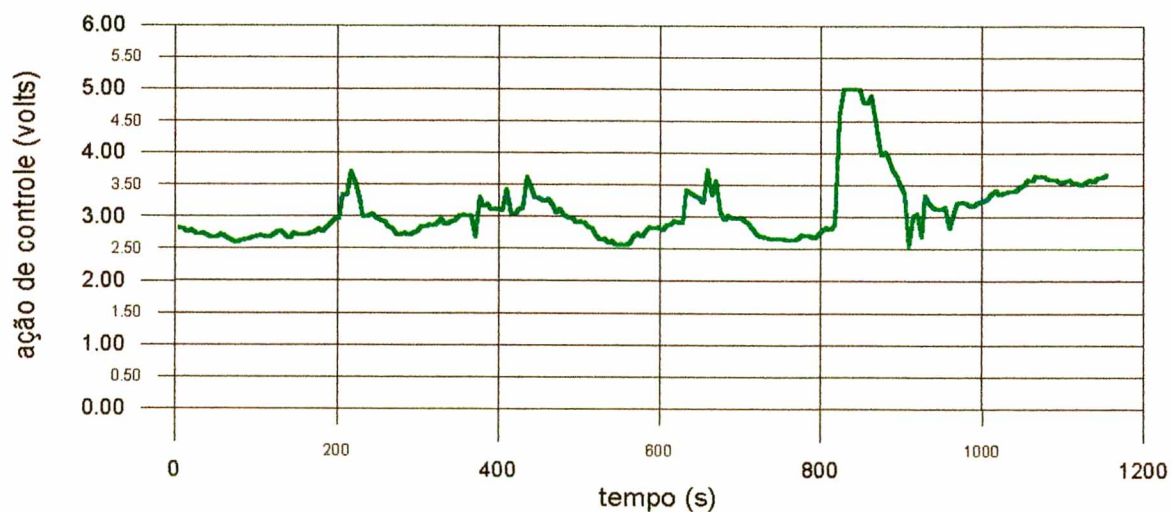
A ação de controle também é calculada pela equação III.42. O controlador utilizado neste sistema é basicamente o mesmo usado para o controle do sistema *tanque simples*, diferindo apenas no modelo que simula a planta. A rede neural utilizada como modelo foi definida no treino realizado para o sistema *tanques acoplados* e cujos parâmetros (pesos) são mostrados nas tabelas V.3 (a) e (b). Utilizou-se $H=1$ e $Nu=1$ e fixou-se o valor de λ em 0,0022 e variou-se α entre 0,55

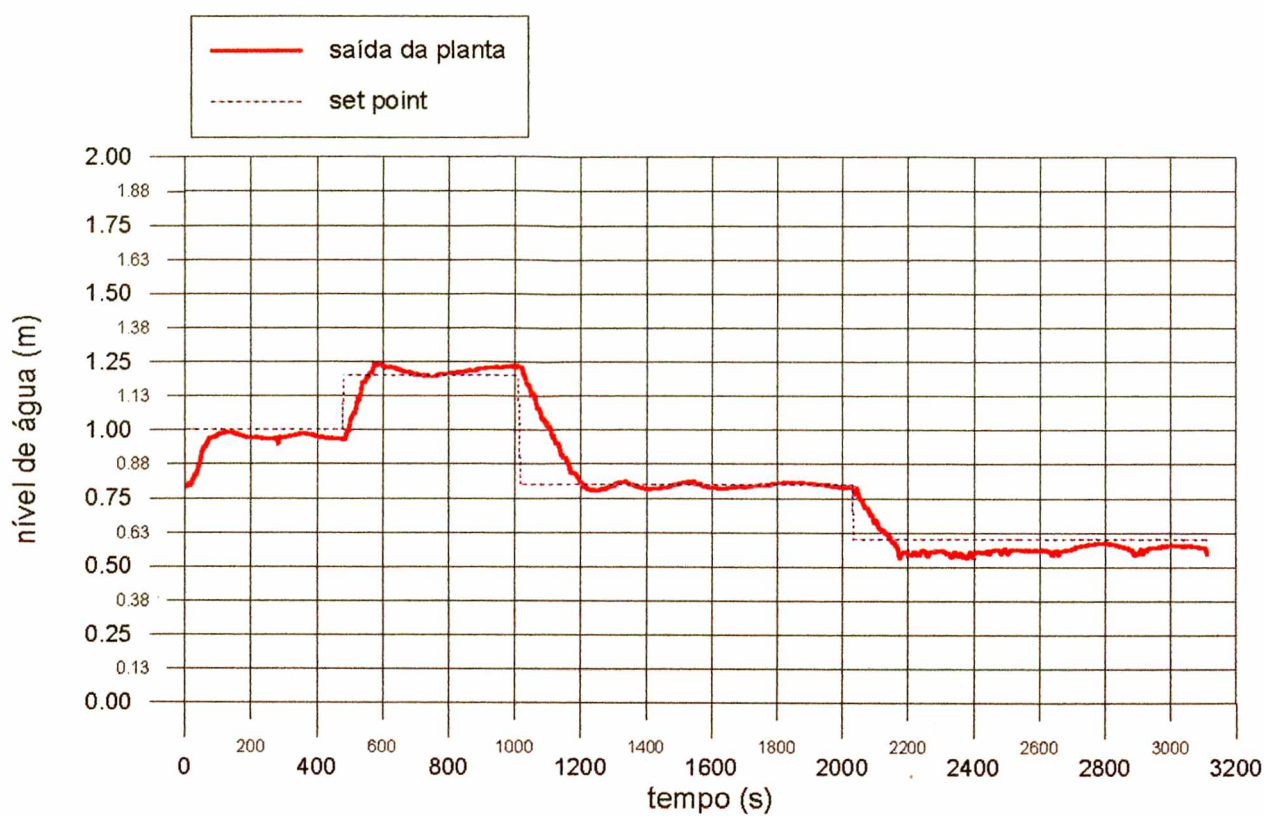
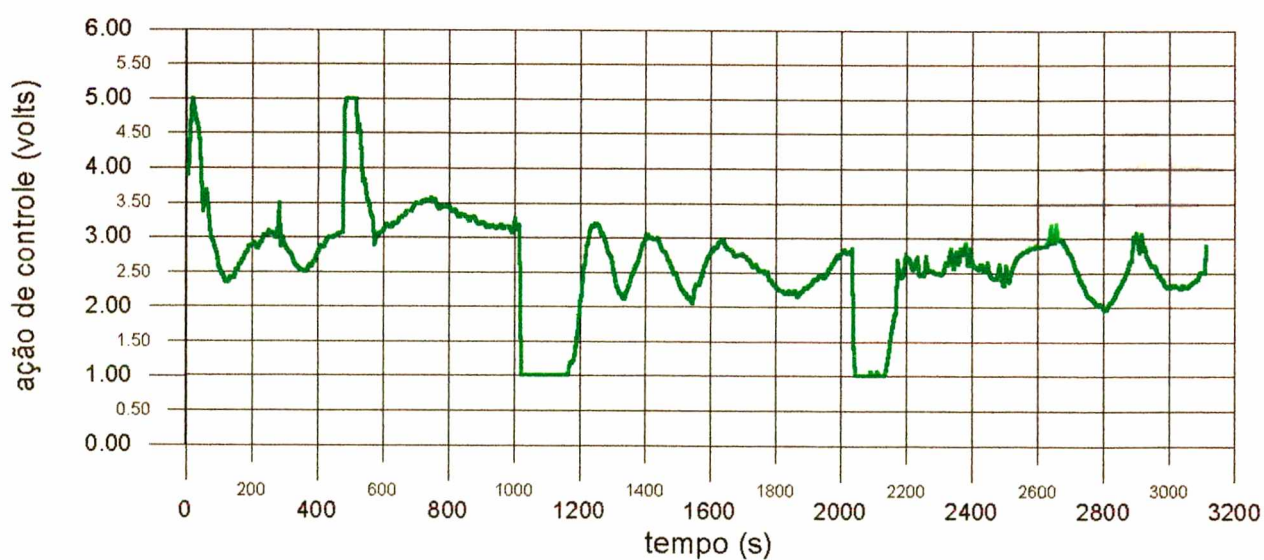
e 0,80 da mesma forma que para o sistema tanque simples. O intervalo de amostragem usado foi de 5 segundos.

Para testar o esquema, diversas mudanças de *set point* e aplicações de carga foram efetuadas. As perturbações na vazão de saída foram geradas pela abertura da válvula VS, de acionamento manual e garantiram aumentos de aproximadamente 10 % e 20 % na vazão. Para este sistema a saída principal também é dada pela válvula manual-VM1, mantida em posição fixa enquanto que a válvula VM2 teve sua posição modificada para gerar cargas de diferentes intensidades. Os resultados do controle são mostrados a seguir:

V.2.3.1 - Comentários

As figuras V.16 (a) e (b) mostram que o controlador proposto foi capaz de manter a variável controlada dentro do valor desejado, bem como fazer a transição do *set point* de 1,00 para 1,25 metros, sem exigir muito da válvula de controle. O mesmo verifica-se nas figuras V.17 (a) e (b). O teste para verificar a capacidade do controlador rejeitar perturbações é mostrado nas figuras V.18 (a) e (b). A carga aplicada no tempo $t = 1610$ segundos e retirada em $t=1750$ segundos representava um aumento de aproximadamente 20% na vazão de saída. Observa-se boa performance do controlador para este caso. Já para a segunda carga, de menor valor (aumento de aproximadamente 10 % na vazão de saída), aplicada em $t = 2700$ segundos e de duração de 200 segundos, o controlador conseguiu fazer com que o sistema voltasse ao ponto de operação desejado sem grandes problemas. Para alguns pontos de operação observa-se uma certa oscilação (amortecida), na ação da válvula de controle. A explicação para isto está no ajuste do controlador, porém o objetivo deste trabalho não é propor uma regra de ajuste ótima e sim verificar a viabilidade do esquema de controle proposto utilizando ANN's. Os próximos resultados mostram esta viabilidade, pois apesar dos problemas ocorridos a rede se mostrou eficiente na modelagem dos sistemas.

Figura V.16 (a) - Mudança no *set point* - sistema *tanques acoplados*Figura V.16 (b) - Ação de controle - Sistema *tanques acoplados*

Figura V.17 (a) - Mudança no *set point* - sistema *tanques acoplados*Figura V.17 (b) - Ação de controle - Sistema *tanques acoplados*

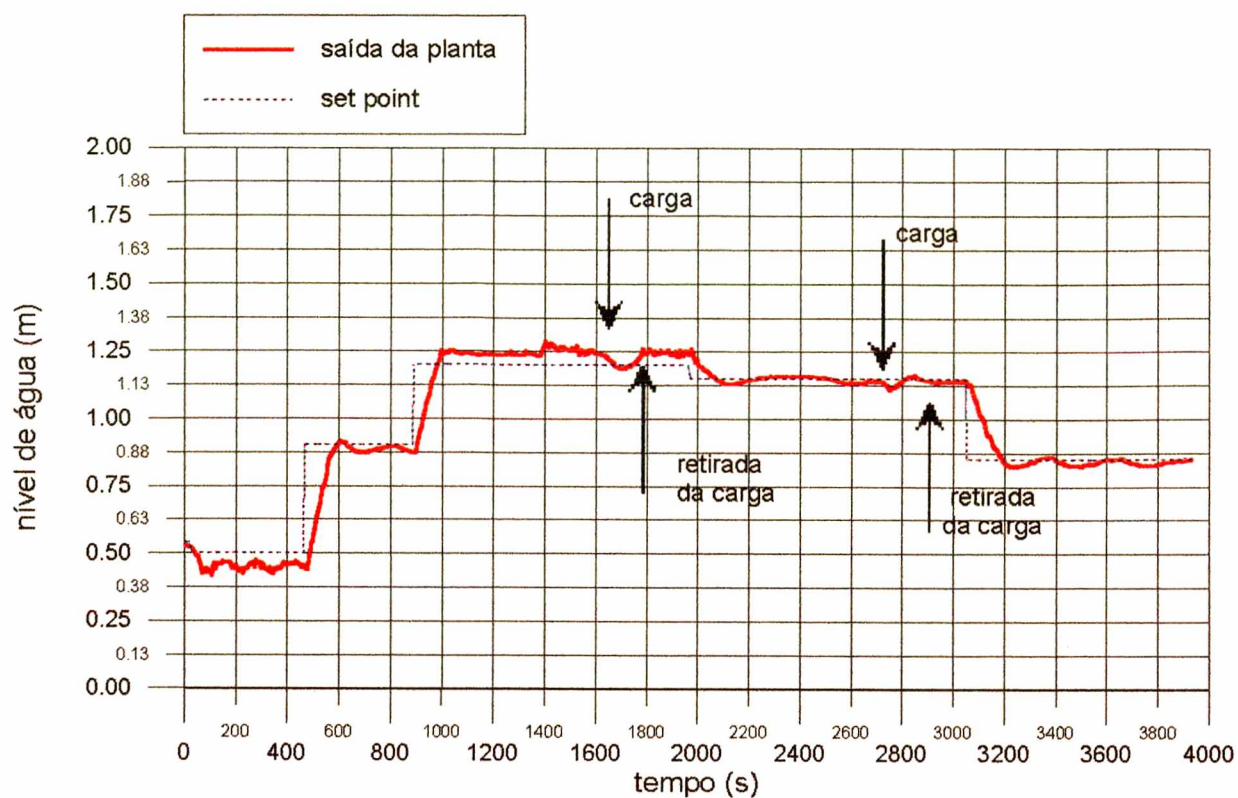


Figura V.18 (a) - Mudança no *set point* e aplicação de carga - sistema *tanques acoplados*

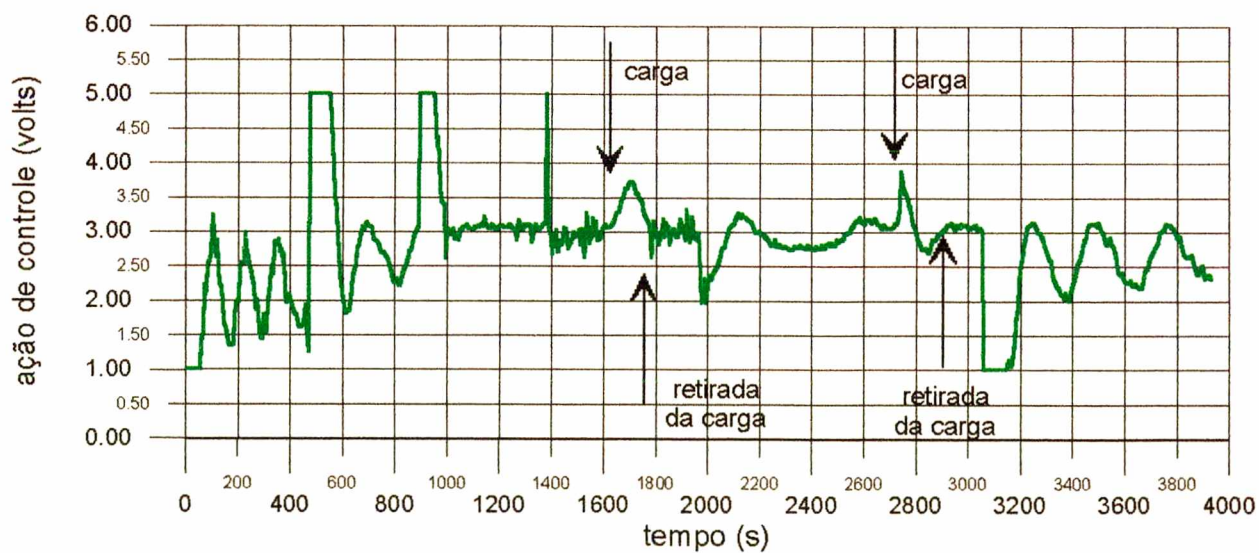


Figura V.18 (b) - Ação de controle - Sistema *tanques acoplados*

V.2.4 - Comportamento da Rede Neural durante o controle do processo

As curvas correspondentes à saída da rede e da planta e o erro absoluto entre ambas são mostradas nas figuras seguintes. A partir destes dados pode-se avaliar o comportamento da rede em laço fechado, para entradas diferentes das quais foi treinada e já atuando *on line*.

V.2.4.1 - Comentários

Observa-se que, em todos os testes realizados para este sistema, a saída da rede acompanha com boa precisão os pontos experimentais. Para os pontos onde o sistema foi perturbado, a rede, apesar de ter sido treinada *off line* em condições diferentes, foi capaz de prever as saídas da planta com exatidão.

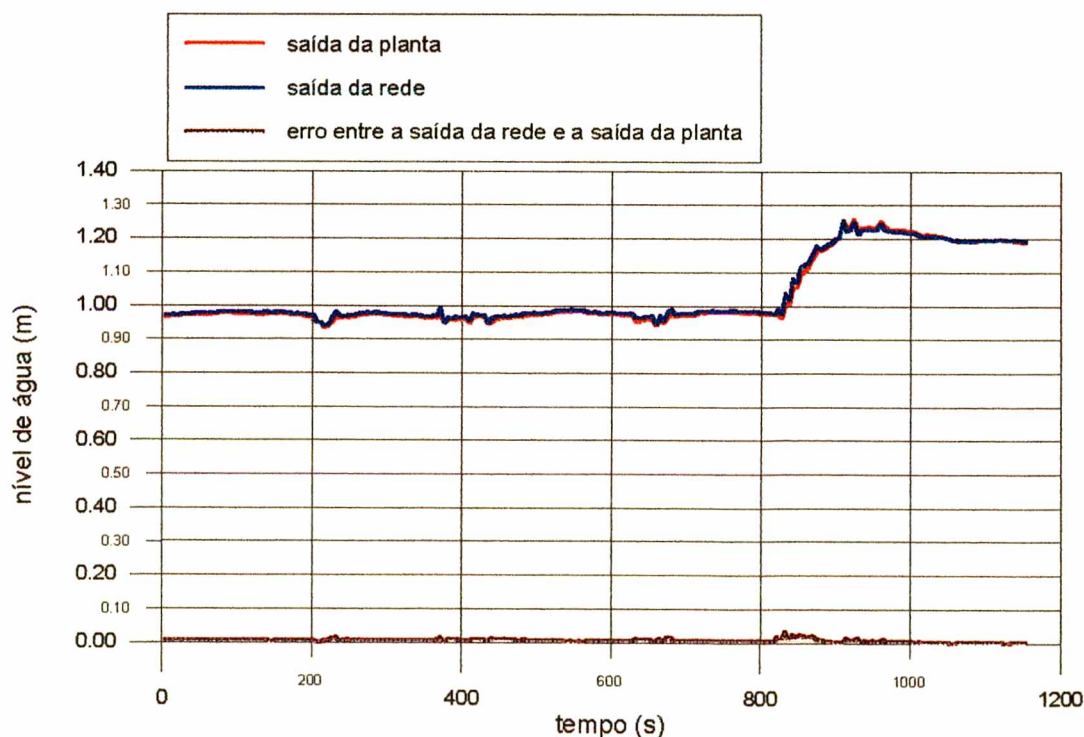
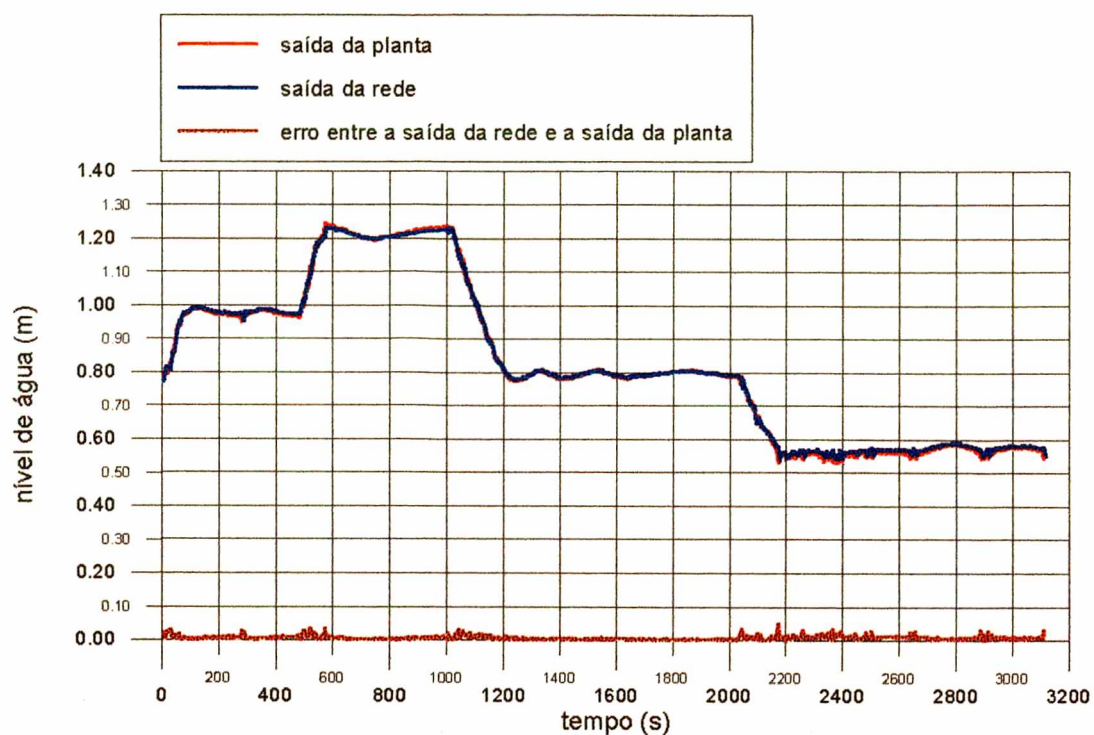
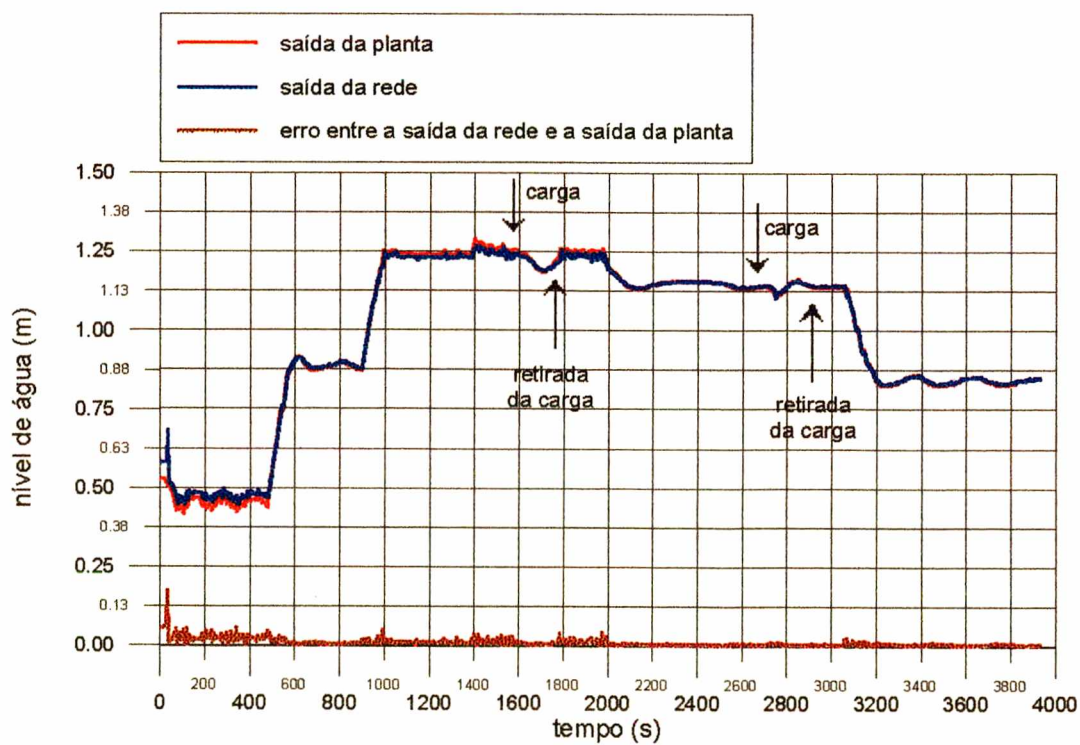


Figura V.19 - Comparação entre a saída da rede e a da planta - sistema *tanques acoplados*

Figura V.20 - Comparação entre a saída da rede e a da planta - sistema *tanques acoplados*Figura V.21 - Comparação entre a saída da rede e a da planta - sistema *tanques acoplados*

CAPÍTULO VI

CONCLUSÕES E SUGESTÕES

Neste capítulo apresenta-se as conclusões referentes aos resultados deste trabalho, bem como sugestões para continuidade.

VI.1 Conclusões

O método de modelagem de processos utilizando redes neurais artificiais mostrou-se eficiente na modelagem dos sistemas *tanque simples* e *tanques acoplados* descritos neste trabalho. Os resultados dos treinos e testes realizados, para a modelagem dos processos, apresentaram desvios pequenos entre a saída do modelo (rede neural artificial) e a saída dos sistemas. Estes desvios foram mais acentuados nos treinos do que nos testes, pois os parâmetros da rede ou pesos foram ajustados durante o treinamento e mantidos fixos nos testes. Os pesos obtidos para a rede que simula o sistema *tanque simples* e para a que simula o sistema *tanques acoplados* são bastante semelhantes, embora os sistemas sejam diferentes em ordem, sendo respectivamente de primeira e segunda ordem. Isto evidencia a capacidade da rede de absorver as características básicas dos sistemas, pois ambos, apesar da diferença de ordem, possuem dinâmica semelhante.

A estratégia de controle proposta, que utiliza redes neurais artificiais como modelo interno no projeto de um controlador preditivo, mostrou-se eficiente no controle do nível dos sistemas *tanque simples* e *tanques acoplados* que formam a unidade experimental. O controlador, do tipo imediato e baseado na minimização de uma função custo, conseguiu efetuar as transições de *set point* e também rejeitar as perturbações impostas.

Para ambos os sistemas, a comparação entre a saída da rede e a saída da planta em malha fechada mostrou que o modelo gerado pela rede neural artificial simula os sistemas apresentando erros pequenos, que podem ser desconsiderados frente aos erros ocasionados por ruídos nas medidas, evidenciando a potencialidade do uso de ANN's na modelagem destes tipos de processos.

V.2 - Sugestões para Futuros Trabalhos

Como continuidade deste trabalho poderia ser adotado um esquema adaptativo de modelagem e controle de processos, onde os pesos da rede usada como modelo interno no projeto do controlador seriam ajustados a cada intervalo de amostragem ou após um determinado período de tempo previamente definido, tornando o controlador, adaptativo.

Poderia-se tentar calcular a saída do modelo neural para H intervalos de amostragem no futuro, onde H , horizonte de predição, seria maior que 1 (um), ou ainda, propor uma regra de ajuste, baseada em uma segunda rede neural, para a identificação dos parâmetros do controlador.

APÊNDICE

1 - Válvula de controle

Marca Hiter - Série 201, atuador DN0021-AC

Ar-abre, igual percentagem, $C_v = 3,5$

Alimentação 20 psi, sinal de comando de 3 a 15 psi.

Parte interna de aço inox 316, corpo de aço carbono.

Conexões tipo rosca de 3/4 in.

2 - Conversor eletropneumático

Marca Hélix - tipo P11-1111-.2

Sinal de entrada - 4 a 20 mA

Sinal de saída - 3 a 15 psi

Alimentação - 20 psi

3 - Sensor de Pressão

Marca Contrisul.

Entrada 0 a 400 mBar.

Saída 4 a 20 mA.

Alimentação 20 V - CC.

Corpo de Aço Carbono.

Parte interna de aço inox 316

4 - Filtro regulador de pressão

Marca Hélix - tipo F11-11.

Alimentação 30 psi.

Saída 20 psi.

5 - Bomba centrífuga

Marca Schneider, modelo 02.01

Potência 1/4 CV, RPM 3400

Alimentação 220 V

6 - Microcomputador / controlador

Microcomputador do tipo PC 386 DX/33 MHz.

Marca Dismac.

Co-processador aritmético 8087-II.

Disco rígido de 100 MB, 1 unidade de disco flexível.

7 - Placa Analógico-Digital / Digital-Analógico

Marca Taurus - 10 bits.

Canais de entrada - 8 canais de 1 a 5 volts

Canais de saída - 8 canais de 1 a 5 volts e 2 canais de 4 a 20 mA.

Tempo de conversão mínimo - 6 μ s.

BIBLIOGRAFIA

ANTSAKLIS, P. J., Guest Ed., Special Issue on Neural Networks for in Control Systems, **IEEE Control Syst. Mag.**, v.10, n.3, p.3-87, april, 1990.

BHAT, N. V., et alli, Modeling Chemical Process Systems via Neural Computation, **IEEE Control Syst. Mag.**, v.10, n.3, p.24-29, april, 1990.

BORGES, L. C. & CASTIER, M., Evaluation of the Thermodynamic Models UNIQUAQ and UNIFAC Using Artificial Neural Networks, **Proceedings of the International Joint Conference on Neural Networks**, Nagoya, 1993 apud DE SOUZA JR. M. B., **Redes Neurais Multicamada Aplicadas a Modelagem e Controle de Processos**, Rio de Janeiro: UFRJ, 1993, p.124, Dissertação de Doutorado, COPPE/PEQ, Universidade Federal do Rio de Janeiro, 1993.

CARVALHO FILHO, E. C. B., LUDERMIR, T. B., DE OLIVEIRA JR, W. R., Apostila do Minicurso Modelagem, Aplicações e Implementações de Redes Neurais. In: **Congresso Brasileiro de Redes Neurais**, I, 1994, Itajubá, MG.

CHEN, F., Backpropagation Neural Networks for Nonlinear Self-tuning Adaptive Control, **IEEE Control Syst. Mag.**, v.10, n.3, p.44-48, 1990.

CHEN, Q. & WEIGAND, W. A., Dynamic Optimization of Nonlinear Processes by Combining Neural Net Model with UDMC, **AIChE Journal**, v.40, n.9, p.1488-1497, September, 1994.

- CHENG, Y., KARJALA, T. W., HIMMELBLAU, D. M., Identification of Nonlinear Dynamic Processes with Unknown and Variable Dead Time Using an Internal Recurrent Neural Network, **Ind. Eng. Chem. Res.**, v.34, n.5, p.1735-1742, 1995.
- CLARKE, D. W., & GAWTHROP, P. J., Self-Tuning Controller, **Proc. IEEE**, v.126, n.6, p.633-640, 1979.
- CLARKE, D. W., MOHTADI, C. & TUFFS, P. S., Generalized Predictive Control - Part-I Basic Algorithm, **Automatica**, v.23, n.2, p.137-160, 1987.
- CLIFFORD, G. Y. L., Guest Ed, Special Issue on Neural Networks I: Theory & Modeling, **Proc. of the IEEE**, v.78, n.9, p 1411-1544, september, 1990.
- DAYAL, B. S., TAYLOR, MACGREGOR, J. F., The Design of Experiments, Training and Implementation of Nonlinear Controllers Based on Neural Networks, **The Canadian Journal of Chem. Eng.**, v.72, p.1066-1079, december, 1994.
- DE SOUZA JR. M. B., **Redes Neurais Multicamada Aplicadas a Modelagem e Controle de Processos**, Rio de Janeiro: UFRJ, 1993, p.124, Dissertação de Doutorado, COPPE/PEQ, Universidade Federal do Rio de Janeiro, 1993.
- EDGAR, T. F. & HIMMELBLAU, D. M., Optimization of Chemical Process, New York, **McGraw-Hill Book Company**, p.651, 1988.
- FAN, J. Y., NIKOLAU, M. & WHITE, R. E., An Approach to Fault Diagnosis of Chemical Processes via Neural Networks, **AIChE Journal**, v.39, n.1, p.82-88, january, 1993.

- GOMIDE, F. A. C., & GUDWIN, R. R., Modelagem, Controle, Sistemas e Lógica Fuzzy, **Controle & Automação**, v.4, n.3, p.97-115, 1994.
- GUPTA, G. & NARASIMHAN, S., Application of Neural Networks for Gross Error Detection, **Ind. Eng. Chem. Res.**, v.32, p.1651-1657, 1993.
- GROSSEBERG, S., Adaptive Pattern Classification and Universal Recoding: I Parallel Development and Coding of Neural Feature Detectors, **Biological Cybernetics**, v.23, 121-134, 1976 apud CLIFFORD, G. Y. L., Guest Ed, Special Issue on Neural Networks I: Theory & Modeling, **Proc. of the IEEE**, v.78, n.9, p 1411-1544, september, 1990.
- HECHT-NIELSEN, R., Theory of the Backpropagation Neural Network, **IEEE Int. Conf. on Neural Networks**, v.I, p.593-605, San Diego, 1989 apud DE SOUZA JR. M. B., **Redes Neurais Multicamada Aplicadas a Modelagem e Controle de Processos**, Rio de Janeiro: UFRJ, 1993, p.124, Dissertação de Doutorado, COPPE/PEQ, Universidade Federal do Rio de Janeiro, 1993.
- HEBB, D. O., The Organization of Behavior, **Wiley**, 1949 apud CARVALHO FILHO, E. C. B., LUDERMIR, T. B., DE OLIVEIRA JR, W. R., **Apostila do Minicurso Modelagem, Aplicações e Implementações de Redes Neurais**. In: Congresso Brasileiro de Redes Neurais, I, 1994, Itajubá, MG.
- HOPFIELD, J. J., Neural Networks and Physical Systems with Emergent Collective Computational Abilities, **Proc. of the National Academy of Sciencies, USA**, v.79, p2554-2558, 1982 apud CLIFFORD, G. Y. L., Guest Ed, Special Issue on Neural Networks I: Theory & Modeling, **Proc. of the IEEE**, v.78, n.9, p 1411-1544, september, 1990.

- LEE, M. & PARK, S., A New Scheme Combining Neural Feedforward Control with Model-Predictive Control, **AIChE Journal**, v.38, n.2, p.193-200, 1992.
- LIMA VERDE, J., R., GUARDANI, R., NASCIMENTO, C. A. O., Modelos de Redes Neurais na Análise de Escoamento em Leito Fluidizado Circulante: Previsão de Perfis de Escoamento de Sólidos, **Resumos do XX Enemp**, São Carlos, 1992 apud DE SOUZA JR. M. B., **Redes Neurais Multicamada Aplicadas a Modelagem e Controle de Processos**, Rio de Janeiro: UFRJ, 1993, p.124, Dissertação de Doutorado, COPPE/PEQ, Universidade Federal do Rio de Janeiro, 1993.
- MCCULLOCH, W. S. & PITTS, W. H., A Logical Calculus of the ideas Immanent in Nervous Activity, **Buie Math Biophys**, v.5, 115-133, 1943 apud CARVALHO FILHO, E. C. B., LUDERMIR, T. B., DE OLIVEIRA JR, W. R., **Apostila do Minicurso Modelagem, Aplicações e Implementações de Redes Neurais**. In: Congresso Brasileiro de Redes Neurais, I, 1994, Itajubá, MG.
- MEGAN, L. & COOPER, D. J., Neural Network Based Adaptive Control via Temporal Pattern Recognition, **The Canadian Journal of Chem. Eng.**, v.70, p.1208-1219, 1992.
- MINSKY, M. L. & PAPERT, S. A. M., Perceptrons, Cambridge, **The Mit Press**, 1969 apud CLIFFORD, G. Y. L., Guest Ed, Special Issue on Neural Networks I: Theory & Modeling, **Proc. of the IEEE**, v.78, n.9, p.1411-1544, september, 1990.
- NGUYEN, D. H. & WIDROW, B., Neural Networks for Sel-Learning Control Systems, **IEEE Control Syst. Mag.**, v.10, n.3, p.18-23, 1990.

- OLIVEIRA, S. R. J. & TEIXEIRA, E. P., Implementação em Laboratório de um Estimador de sistemas Dinâmicos Não Lineares usando Redes Neurais Artificiais. In: Congresso Brasileiro de Redes Neurais, I, 1994, Itajubá. **Anais...** MG: 1994, p.241-246.
- PSICHOGIOS, D. C. & UNGAR, L. H., Direct and Indirect Model Based Control Using Artificial Neural Networks, **Ind. Eng. Chem. Res.**, v.30, n.12, p.2564-2573, 1991.
- RUMELHART, D. E. & McCLELLAND, J. L., Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Cambridge, **Mit Press**, 1986, vol.1 apud DE SOUZA JR. M. B., **Redes Neurais Multicamada Aplicadas a Modelagem e Controle de Processos**, Rio de Janeiro: UFRJ, 1993, p.124, Dissertação de Doutorado, COPPE/PEQ, Universidade Federal do Rio de Janeiro, 1993.
- THOMPSON, M. L. & KRAMER, M. A., Modeling Chemical Processes Using Prior Knowledge and Neural Networks, **AIChE Journal**, v.40, n.8. p.1328-1340, August, 1994.
- VENKATASUBRAMANIAN, V. & CHAN, K., A Neural Network Methodology for Process Fault Diagnosis, **AIChE Journal**, v.35.n.12, p.1993-2002, 1989.
- WATANABE, K. et alli, Incipient Fault Diagnosis of Chemical Process via Artificial Neural Networks, **AIChE Journal**, v.35, n.11, p.1803-1812, 1989.
- WATANABE, K. HIROTA, S., HOU, L., Diagnosis of Multiple Simultaneous Fault via Hierarchical Artificial Neural Networks, **AIChE Journal**, v.40, n.5, p.839-848, May, 1994.